

01 | 2015

KOS.content

Ergebnisse der Untersuchungen des
Kompetenzzentrum Open Source der DHBW-Stuttgart

Frühjahr 2015
band.1

INHALT BAND.1

Inhalt __

Das Kompetenzzentrum Open Source (KOS)

Ziel des Projektes

Das Projekt Kompetenzzentrum Open Source der DHBW Stuttgart wurde mit der Zielsetzung ins Leben gerufen, die Einsatzfelder für Open Source Software in Unternehmen zu identifizieren und durch den Einsatz quelloffener Produkte und deren kostengünstigen Einsatzmöglichkeiten Optimierungen in ausgewählten Geschäftsbereichen zu erzielen.

Dies bedeutet konkret, dass z.B. Open Source Software evaluiert wird, um Lizenzkosten zu reduzieren, bewertet wird, ob sie diverse Qualitätskriterien erfüllt und erfolgreich(er) und effizient(er) in Unternehmen genutzt werden kann. Das Ziel des Projektes ist es hierbei, allgemeingültige Lösungskonzepte für Problemstellungen zu erarbeiten, welche von den am Projekt beteiligten Unternehmen zu firmenspezifischen Lösungen weiterentwickelt werden können. Die beteiligten Unternehmen partizipieren so an den Ergebnissen des Projekts.

Zusammenarbeit mit den Dualen Partnern

Die Zusammenarbeit mit den Dualen Partnern gestaltet sich entlang deren Anforderungen und Bedürfnissen. Sie sind die Themengeber für betriebliche Fragestellungen, die im Rahmen des Projekts untersucht werden. Die DHBW steuert die wissenschaftliche, didaktische und methodische Expertise und Forschungskompetenz bei und untersucht die identifizierten Themenfelder.



Im Rahmen des Projektes steuert die DHBW Stuttgart die wissenschaftliche Expertise und Forschungskompetenz bei zur Bearbeitung der betrieblichen Fragestellungen der Dualen Partner. Es entstehen generische Lösungen, welche von den Partnern an Ihre Situation angepasst werden kann.
Im Rahmen der Arbeit entstehen (generische) Lösungen, an denen die Partner teilhaben können indem sie diese auf ihre spezifische Unternehmenssituation anpassen. Zudem fließen die Ergebnisse in die Arbeit der DHBW ein, sodass hier dem Anspruch an eine hohe Anwendungs- und Transferorientierung ganz im Sinne einer kooperativen Forschung Rechnung getragen wird.

An den Ergebnissen des Projekts partizipieren die Dualen Partner Allianz Deutschland AG, die Deutsche Rentenversicherung Baden-Württemberg, die HALLESCHE Krankenversicherung a.G. und die WGV-Informatik und Media GmbH.



Open Source-Produkte zum Aufbau von Private Clouds

Schriftliche Ausarbeitung
im Rahmen der Lehrveranstaltung „Integrationsseminar“
für das Kompetenzzentrum Open Source (KOS)

Vorgelegt von

Amelie Böer, Thomas Enderle
Matthias Holzwarth, Dominik Schrahe

am 23.01.2015

Fakultät Wirtschaft
Studiengang Wirtschaftsinformatik
WWI2012V

Inhaltsverzeichnis

Abkürzungsverzeichnis

2FA	**Zwei-F**aktor-**A**uthentifizierung
AES	**A**dvanced **E**ncryption **S**tandard
DAX	**D**eutscher **A**ktieninde**x**
DH	**D**iffie-**H**ellman-Schlüsselaustausch
DHBW	**D**uale **H**ochschule **B**aden-**W**ürttemberg
DIN	**D**eutsches **I**nstitut für **N**ormung
ECDH	**E**lliptic **C**urve **D**iffie-**H**ellman
ECDHE	**E**lliptic **C**urve **D**iffie-**H**ellman **E**phemeral
FAQ	**F**requently **A**sked **Q**uestions
IaaS	**I**nfrastructure **as a S**ervice
IEEE	**I**nstitute of **E**lectrical and **E**lectronics **E**ngineers
IT	**I**nformations**t**echnik
KOS	Golembowska, A. u. a. (2012)
MITM	**M**an-**i**n-**t**he-**M**iddle
NIST	**N**ational **I**nstitute of **S**tandards and **T**echnology
PaaS	**P**latform **as a S**ervice
PFS	**P**erfect **F**orward **S**ecrecy
PIN	**P**ersönliche **I**dentifikations**n**ummer
RSA	**R**ivest, **S**hamir und **A**dleman-Verfahren
SaaS	**S**oftware **as a S**ervice
SHA2	**S**ecure **H**ash **A**lgorithm Version **2**
SOA	**s**ervice-**o**rientierte **A**rchitektur
TAN	**T**rans**a**ktions**n**ummer

Abbildungsverzeichnis

1 Einleitung

Autoren: Amelie Böer, Thomas Enderle, Matthias Holzwarth, Dominik Schrahe

Vor drei Jahren wurde „Cloud Computing“ als Trend beschrieben, mit dessen Hilfe IT-Ressourcen bedarfsgerecht mithilfe des Internets ausgelagert werden kann. Nachdem diese Technologie mittlerweile als etabliert bezeichnet werden darf, stehen vermehrt Überlegungen von verschiedenen Unternehmen an, diese einzusetzen. Die abstrahierte Verfügbarkeit über ein Netzwerk bietet dabei Vorteile und Gefahren, weshalb eine genaue Betrachtung dieser Thematik eine Grundlage für weiterführende Überlegungen bezüglich des unternehmensinternen Einsatzes ist.

Im Rahmen des Moduls „Projekt“ im fünften Semester der Dualen Hochschule Baden-Württemberg wird diese Arbeit von vier Studenten angefertigt. Durch ihre Tätigkeit in der Softwareentwicklung und der Betriebsorganisation in verschiedenen Versicherungshäusern sind sie mit der oben beschriebenen Thematik vertraut.

Ziel dieser Ausarbeitung ist es, dem Leser einen Überblick über die verschiedenen auf dem Markt vorhandenen Cloud-Realisationen zu geben. Zudem wurde, als Produkt dieser Arbeit, ein Kriterienkatalog entwickelt. Dieser kann von interessierten Unternehmen ausgefüllt werden.

Um mit dieser Arbeit dem Auftraggeber einen speziellen Mehrwert zu bieten, ist ein Großteil der theoretischen und praktischen Überlegungen auf die Begebenheiten in Versicherungshäusern zugeschnitten. Dies ist von Belang, da gerade in dieser Branche wird stetig mit sensiblen Daten gearbeitet, weshalb eine genaue Betrachtung der Einsatzmöglichkeiten von Clouds und eine sorgfältige Produktauswahl hier von besonderer Bedeutung sind. Da diesen Anforderungen begegnet wurde, ist diese Ausarbeitung besonders für IT-Mitarbeiter in verschiedenen Versicherungshäusern zur Lektüre empfohlen.

Zunächst wird in Kapitel 2 der Begriff der Cloud geklärt. Hier werden dem Leser auch verschiedene Unterteilungsarten von Clouds dargelegt. Hierbei wird auf die technische und organisatorische Differenzierung eingegangen. Für diese Ausarbeitung des Forschungsbereichs der Dualen Hochschule Baden-Württemberg existieren spezielle Begebenheiten auf welche die gesamte Arbeit ausgerichtet ist. Um diesem gerecht zu werden wird zunächst Open Source beschrieben. Die Notwendigkeit für diese Beschreibung besteht, da eine Konkrete Vorgabe, vor allem wegen der Zusammenarbeit mit dem Kompetenzzentrum Open Source der DHBW. Da für dieses Projekt ein Versicherungshaus Partner ist, wird speziell auf die Begebenheiten eingegangen, die sich in dieser Branche finden. Auf diese wird dann im weiteren Verlauf der Ausarbeitung eingegangen.

In Teil 4 folgt eine Beschreibung, warum Private Clouds zur Anwendung kommen müssen, speziell wird hierbei auf die Thematik des Datenschutzes eingegangen, der vor allem auch für Versicherungen relevant ist. Eine konkrete Vorstellung von ausgewählten Produkten findet im folgenden Teil 5 statt.

Die theoretischen Grundlagen der Thematik der Sicherheit folgen am anschließenden Kapitel. Dies ist notwendig, da die Nutzung von Cloud Computing nur Angedacht werden darf, sofern die nötigen Sicherheitsüberlegungen getätigt und umgesetzt wurden. Speziell eingegangen wird hierbei auf Authentifizierung, Verschlüsselung und Integrität.

Der 7. Teil beschäftigt sich anschließend mit dem Kriterienkatalog, welcher von den Autoren erstellt wurde, um eine Cloud-Lösung auszuwählen. Begonnen wird damit, dass zunächst das gewählte Vorgehen für die Erstellung des Kriterienkatalogs beschrieben und abgebildet wird. Anschließend folgt eine wissenschaftliche Begründung der Auswahl dieser. Die konkrete Anwendung des Kataloges folgt in Teil 8, die Ergebnisse werden auch miteinander verglichen.

Um einen Praxisbezug herzustellen wird im 0. Kapitel eine exemplarische Realisierung einer Private-Cloud-Lösung am Beispiel des Produkts ownCloud dargestellt, welches mit Screenshots untermalt wird. Auch praktische Erfahrungen werden für den Leser festgehalten.

Abschließend wird, im finalen Kapitel 10 ein Fazit und Ausblick gegeben.

2 Clouds

Autoren: Amelie Böer, Thomas Enderle, Matthias Holzwarth, Dominik Schrahe

2.1 Grundlegendes

In dieser Arbeit wird eine Cloud wie folgt definiert: Die Bereitstellung und Nutzung von IT-Infrastruktur, Plattformen und Anwendungen in einem Netzwerk. Sie ist eine Dienstleistung eines Cloud-Anbieters, welche einem oder mehreren Cloud-Nutzern zur Verfügung gestellt wird. Darüber hinaus hat das National Institute of Standards and Technology (NIST) einen Kriterienkatalog zur Beschreibung von Clouds erstellt. Abb. 1 visualisiert diesen Kriterienkatalog (**Essential Characteristics**), drei Dienstklassen (**Service Models**), sowie vier Betriebsmodelle (**Deployment Models**).

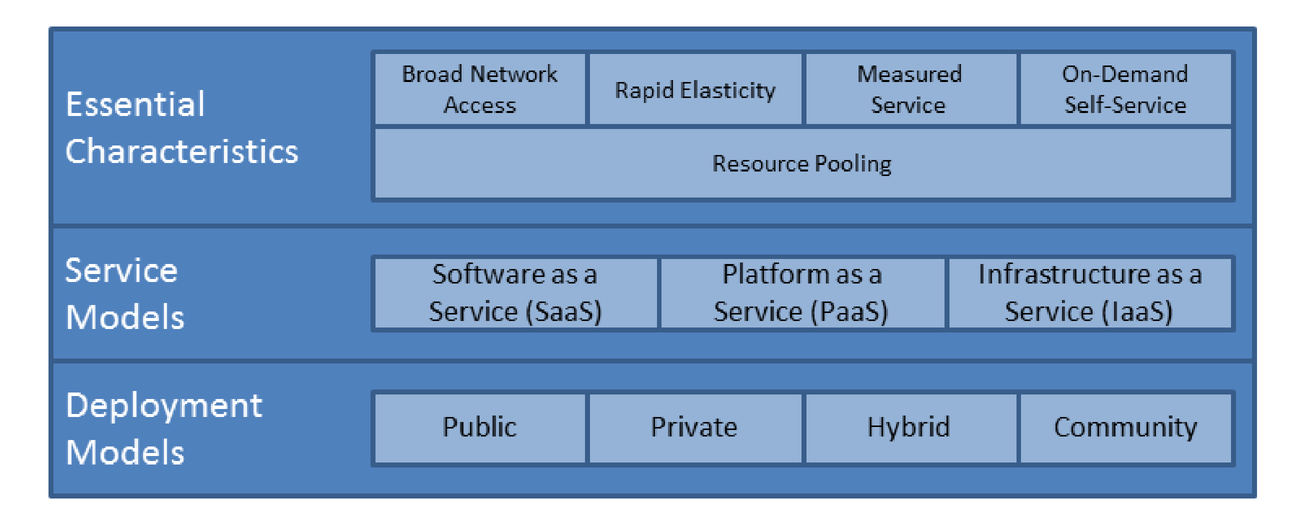


Abb. 1: NIST-Definition für Clouds[1]

Im Folgenden sollen die Essential Characteristics näher beleuchtet werden. Die Service Models können auch als technische Unterteilung von Clouds angesehen werden, während die Deployment Models eine organisatorische Unterteilung darstellen. Auf sie wird in Kapitel 2.2 eingegangen.

Broad Network Access bedeutet, dass Cloud-Dienste netzwerkbasiert angeboten und mit Hilfe von Standard-Technologien ausgeliefert werden.[2] Meist erfolgt die Auslieferung mittels eines Webbrowsers.[3]

On-Demand Self-Service meint, dass der Cloud-Nutzer die Dienstleistung des Cloud-Anbieters bei Bedarf verwendet. Um die Dienstleistung zu nutzen ist keine weitere Interaktion mit dem Cloud-Anbieter nötig.[4] Benötigt der Cloud-Nutzer die Dienstleistung nicht, werden ihm keine Kosten in Rechnung gestellt.[5] Hierdurch werden Fixkosten für Hard- und Software in variable Kosten für den Service umgewandelt werden.

Daraus lässt sich die Eigenschaft **Resource Pooling** ableiten. Resource Pooling bedeutet, dass die Erbringung der Dienstleistung parallel an unterschiedliche Nutzer (auch Mandanten genannt) erfolgt. Die Dienstleistung ist dabei an den momentanen Bedarf eines Nutzers angepasst.[6]

Um für alle Cloud-Nutzer eine gleichbleibende Qualität sicherzustellen, muss die Cloud **Rapid Elasticity**-fähig sein. Hiermit ist die dynamische Skalierung der Dienstleistung bei

[1] Mit Änderungen entnommen aus: Yeluri, R./Castro-Leon, E. (2014), S. 2
[2] Vgl. Baun, C. u. a. (2011), S. 5
[3] Vgl. Eymann, T. (2014)
[4] Vgl. Baun, C. u. a. (2011), S. 5
[5] Vgl. ebenda, S. 6
[6] Vgl. ebenda, S. 5

Bedarf gemeint. Dem Nutzer wird das Gefühl vermittelt, dass ihm beinahe unendlich Ressourcen zur Verfügung stehen, sollte er diese benötigen.[7]

Für den Cloud-Nutzer und den Cloud-Anbieter ist es wichtig, dass die Dienstleistung messbar ist (**Measured Service**). Die Daten werden zum Beispiel für die dynamische Skalierung oder die nutzungsabhängige Bezahlung benötigt.[8]

Sind die erläuterten Bedingungen erfüllt, so handelt es sich bei einer Dienstleistung um eine Cloud.

Im Kontext von Clouds existieren verschiedene Technologien, die als Basis zur Implementierung von Clouds dienen. Diese werden in diesem Abschnitt vorgestellt und erläutert.

Eine dieser Technologien ist die **Virtualisierung**. Virtualisierung ist die Abstraktion einer physischen Ressource, wie zum Beispiel Servern. Auf diese Weise können auf einem physischen Server mehrere virtuelle Server betrieben werden. Der Vorteil ist, dass sich eine günstigere Auslastung erzeugen lässt. Hierdurch ergibt sich ein Kostenvorteil. Es müssen weniger physische Ressourcen zur Verfügung stehen um beispielsweise Lastspitzen auszugleichen. Ein weiterer Vorteil der Virtualisierung ist die automatisierte Erstellung von virtuellen Maschinen, wodurch sich eine Rapid Elasticity kostengünstig implementieren lässt.[9]

Auch für den Kunden ergeben sich durch die Virtualisierung Vorteile: Die Dienste des Cloud-Anbieters können hoch verfügbar gehalten werden. Durch Aktualisierungen der Technologie ergeben sich keine Wartungsfenster, da die virtuellen Maschinen problemlos zwischen physischen Maschinen verschoben werden können. Zudem sind virtuellen Maschinen isoliert, sowohl untereinander als auch von der physischen Komponente. Daraus ergeben sich hauptsächlich folgende Vorteile:

- Multimandantenfähigkeit (verschiedene Kunden arbeiten auf einer Instanz[10])
- Abgabe von Managementfunktionen an den Kunden
- Verbesserung der Sicherheit

Trotz der Vorteile besitzt die Virtualisierung einen entscheidenden Nachteil: Die Virtualisierung selbst benötigt Rechenleistung. Dieser Umstand ist heutzutage, durch aktuelle und moderne Virtualisierungstechniken, kaum relevant. Zudem überwiegen die Vorteile der Virtualisierung ihre Nachteile.[11]

[7] Vgl. Baun, C. u. a. (2011), S. 6
[8] Vgl. Yeluri, R./Castro-Leon, E. (2014), S. 3
[9] Vgl. Baun, C. u. a. (2011), S. 9 f.
[10] Vgl. Böhm, M. u. a. (2010), S. 97
[11] Vgl. Baun, C. u. a. (2011), S. 11 f.

Eine weitere grundlegende Technologie des Cloud Computings ist die service-orientierte Architektur (SOA). Service-Orientierung ist ein Konzept um Applikationen und Methoden als IT-Services anzubieten oder in Enterprise-Anwendungen einzubinden. Hierdurch ist eine plattform- und sprachenunabhängige Nutzung der Services möglich.[12]

Grundlage einer SOA sind verteilte Systeme, die über ein Netzwerk miteinander kommunizieren.[13] In einer SOA sind die Services lose gekoppelt. Lose Kopplung bedeutet, dass die einzelnen Services keine oder nur geringe Abhängigkeiten vorweisen.[14] Die Services kommunizieren nicht direkt über (proprietäre) Protokolle miteinander[15], sondern über eine Middleware mit Hilfe von Standard-Protokollen wie XML.[16]

Die Vorteile sind eine hohe Flexibilität bei Änderungen durch die lose Kopplung und eine bessere Wartbarkeit durch Wiederverwendung bereits bestehender Services.[17] Zudem wird die Skalierbarkeit verbessert.[18]

2.2 Unterteilung von Clouds

Die Unterteilung von Clouds ist sowohl nach technischen als auch nach organisatorischen Gesichtspunkten möglich. Im Folgenden soll auf beide eingegangen werden.

Technische Unterteilung

Aus technischer Sichtweise sind bei Cloud Computing-Lösungen grundsätzlich drei Ansätze unterscheidbar. Diese erfüllen unterschiedliche Unternehmensanforderungen. Wünscht sich ein Unternehmen möglichst hohe Gestaltungsfreiheit bis auf Betriebssystemebene, bietet sich Infrastructure as a Service (IaaS) als Lösung an. Hierbei wird die Hardware bereitgestellt. Dies umfasst persistenten sowie flüchtigen Speicher, Rechenleistung und Netzwerkanbindung. Betriebssysteminstanzen können dagegen in der Regel vom Benutzer selbst angelegt und ausgeführt werden. Insbesondere im IaaS-Bereich nehmen Virtualisierungstechnologien eine wichtige Position ein. Da einzelne Betriebssysteminstanzen weiterhin auf physischer Hardware ausgeführt werden können sind derartige Technologien potentiell nicht notwendig, die meisten IaaS-Produkte arbeiten jedoch damit. Beispiele für diesen Bereich stellen unter anderem Amazons Elastic Compute Cloud für Berechnungen sowie der Simple Storage Service zur Datenablage dar. Darüber hinaus werden im Private Cloud-Umfeld die Produkte CloudStack, Eucalyptus und OwnCloud den IaaS-Lösungen zugeordnet.[19]

[12] Vgl. Finger, P./Zeppenfeld, K. (2009), S. 3 ff.
[13] Vgl. Schill, A./Springer, T. (2012), S. 3
[14] Vgl. Finger, P./Zeppenfeld, K. (2009), S. 7
[15] Vgl. ebenda, S. 8
[16] Vgl. ebenda, S. 6 f.
[17] Vgl. ebenda, S. 7
[18] Vgl. ebenda, S. 8
[19] Vgl. Baun, C. u. a. (2011), S. 29 ff.

Falls die Anforderungen über die reine Bereitstellung der Hardware hinausgehen stellen Platform as a Service (PaaS) Produkte möglicherweise die bessere Lösung dar. Die Zielgruppe hierfür sind vor allem Softwareentwickler, die Entwicklungs- oder Laufzeitumgebungen für ihre Arbeit benötigen, sich jedoch nicht mit der Hardwareinfrastruktur darunter beschäftigen möchten.

Als Beispiel lässt sich in diesem Bereich Microsoft Azure anführen.[20] Neben diesem Public Cloud-Angebot existieren auch Open Source IaaS-Lösungen im Private Cloud-Bereich wie AppScale.

Neben Iaas und PaaS stellen Software as a Service (SaaS) Lösungen einen wichtigen Bereich dar. Kunden müssen hierbei keine Software mehr auf Clients installieren, sondern zahlen lediglich für den Zugriff auf die vom Cloud-Privider bereitgestellten Anwendungen. Diese reichen von der relativ einfachen Dokumentenbearbeitung bei Google Docs bis hin zu komplexen CRM-Lösungen wie sie Salesforce bietet.[21]

Organisatorische Unterteilung

Aus organisatorischer Sicht sind vier Arten von Clouds zu unterscheiden.

Um die Unterschiede zu verdeutlichen, bietet es sich an wie in Abb. 2 dargestellt zwischen drei unterschiedlichen Übertragungswegen zu differenzieren. Unter lokal versteht man hierbei lokal beim Unternehmen. Unter dediziert wird in diesem Zusammenhang eine, zum Beispiel von einem Dienstleister, ausschließlich für die Anwendung zur Verfügung gestellte Netzwerkinfrastruktur verstanden.[22] Ein öffentlicher Zugang wäre beispielsweise das Internet.

[20] Vgl. Baun, C. u. a. (2011), S. 35 f.
[21] Vgl. ebenda, S. 37 f.
[22] Vgl. Matros, R. (2012), S. 65

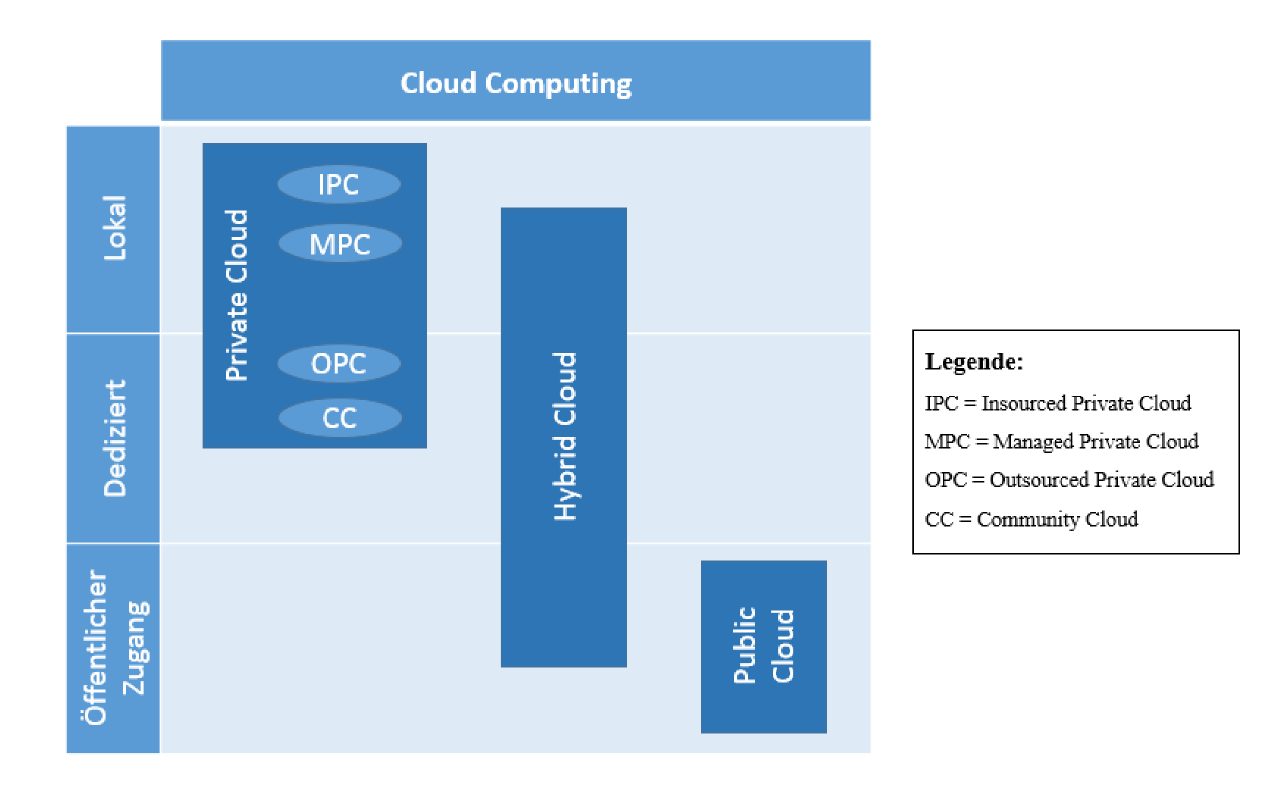


Abb. 2: Organisatorische Unterteilung von Clouds[23]

Im Falle der Public Cloud befindet sich diese im Eigentum eines Dienstleisters und wird von ihm betrieben.[24] Die Leistung wird über einen öffentlichen Zugang erbracht und entspricht dem one-to-many-Ansatz.[25] Anbieter und potentiellen Nutzer gehören hierbei nicht derselben organisatorischen Einheit an. Der Nutzer kann sich den gewünschten Leistungsumfang selbst spezifizieren. Hierfür ist meist kein übergeordneter Rahmenvertrag notwendig, da die vertragliche Bindung im Rahmen der Leistungsspezifikation getroffen wird.[26] Bei einer Public Cloud ist aufgrund der Ausrichtung auf die hohe Kundenanzahl der Automatisierungsgrad von besonderer Bedeutung. Sie müssen daher standardisiert und elastisch[27] sein.[28]

Im Gegensatz zur Public Cloud folgt die Private Cloud dem one-to-one-Ansatz. Sie im lokalen und dedizierten Bereich anzufinden und keiner breiten Kundengruppe zugänglich.[29] Der Anbieter und der Nutzer gehören hier für gewöhnlich demselben Unternehmen bzw. derselben organisatorischen Einheit an. Hauptbeweggrund für den Einsatz von Privat Clouds sind Sicherheitsbedenken bzw. der Schutz sensibler Daten. Für die Private Cloud entwickelte Anwendungen werden häufig über Schnittstellen initialisiert, welche auch für Public Clouds angewendet werden. Ziel dieser Herangehensweise ist die Möglichkeit, den Dienst später

[23] Mit Änderungen entnommen aus: Matros, R. (2012), S. 66
[24] Vgl. Barton, T. (2014), S. 45
[25] Vgl. Matros, R. (2012), S. 66
[26] Vgl. Baun, C. u. a. (2011), S. 27 f.
[27] siehe Anhang 1
[28] Vgl. Matros, R. (2012), S. 66
[29] Vgl. ebenda, S. 67

auch in letzterer einsetzen zu können. Zudem können auch in der Public Cloud vorhandene Werkzeuge genutzt werden.[30] Aufgrund der begrenzten Nutzergruppe kann bei einer Private Cloud viel gezielter auf das individuelle Kundenbedürfnis eingegangen werden.[31]

Die Menge der Private Clouds kann zudem in vier Bereiche untergliedert werden:

1. Insourced Private Cloud:
   Diese Art der Private Cloud wird durch das Unternehmen selbst bzw. deren Mitarbeiter betrieben und das Unternehmen bleibt der Eigentümer der Infrastruktur.[32]

2. Managed Private Cloud:
   Auf Grundlage individueller Service-Level-Agreements wird diese Form der Private Cloud von externen Dienstleistern betrieben.[33]

3. Outsourced (oder Hosted) Private Cloud:
   Auch bei dieser Variante wird die Cloud durch den Dienstleister betrieben. Dies erfolgt in eigener Verantwortung. Die Cloud befindet sich allerdings zusätzlich auch physisch beim Dienstleister, welcher der Besitzer der IT-Ressourcen ist.[34]

4. Community Cloud:
   Unter Community Cloud versteht man eine Kopplung von Private Clouds verschiedener Unternehmen oder Organisationen, welche derselben Branche angehören. Sie ist als eine Sonderform der Private Cloud zu verstehen, welche nur den beteiligten Mitgliedern zur Verfügung steht und in dedizierten Netzwerken angesiedelt ist. Community Clouds sind zu empfehlen, wenn eine gemeinsame Nutzung der vorhandenen Ressourcen aufgrund ähnlicher Anforderungen bzw. Aufgaben sinnvoll ist.[35]

Die dritte Art von Clouds bezeichnet man als Hybrid Clouds. Sie sind eine Kombination von Diensten aus Public und aus Private Clouds. Hierbei wird der Regelbetrieb üblicherweise in der Private Cloud abgewickelt. Kommt es zu Lastspitzen, werden diese in der Public Cloud verarbeitet. Zudem können ausgewählte Funktionalitäten ausgelagert werden. Die Thematik des Datenschutzes ist hierbei kritisch zu betrachten.[36] Eine weitere Form der Hybrid Cloud ist die Kombination einer Cloud mit einer traditionellen IT-Landschaft.[37]

[30] Vgl. Baun, C. u. a. (2011), S. 28 f.
[31] Vgl. Matros, R. (2012), S. 67
[32] Vgl. Barton, T. (2014), S. 46
[33] Vgl. ebenda, S. 46
[34] Vgl. ebenda, S. 46
[35] Vgl. ITwissen (o. J.)
[36] Vgl. Baun, C. u. a. (2011), S. 29
[37] Vgl. Barton, T. (2014), S. 46

3 Auftragsbezogene Besonderheiten

Autoren: Amelie Böer, Thomas Enderle, Matthias Holzwarth, Dominik Schrahe

Für die Bearbeitung des Auftrages gibt es zwei Besonderheiten, die es zu beachten gilt. Zum einen müssen die Cloud-Produkte, die durch die Autoren kategorisiert wurden, Open Source-Produkte sein. Deshalb wird im nachfolgenden Abschnitt ein kurzer Überblick über Open Source-Software gegeben.

Zum anderen müssen die Charakteristika der Versicherungsbranche in die Betrachtung einbezogen werden. Beispielsweise die hohen Anforderungen an den Datenschutz und die Datensicherheit. Diese und weitere Merkmale werden im Anschluss an die Open Source-Grundlagen erläutert.

Open Source-Software

Aufgabe dieser Projektarbeit ist es, wie bereits in den einleitenden Kapiteln geschildert, die Einsatzmöglichkeiten von Clouds aus dem Open-Source-Bereich zu analysieren. Dies setzt die Kenntnis der Grundlagen und Vorteile von Open Source voraus, welche in diesem Teil beschrieben werden.

Bei Open Source handelt es sich um Software, deren Quellcode öffentlich einsehbar ist.[38] Die Nutzung und Anpassung des Programms ist dadurch möglich. Der Gebrauch von Open-Source-Software für kommerzielle Zwecke wird dadurch ebenfalls ermöglicht[39], was ihren Einsatz in Versicherungshäusern begünstigt. Das Prinzip der Transparenz wird somit erreicht. Grundlegend für Open Source ist zudem, dass (per Grundverständnis) keine Lizenzgebühren erhoben werden dürfen.[40] Die Weitergabe der Software wird somit vereinfacht. Zudem besteht der Vorteil für das Unternehmen, dass keine Kosten für die Lizenzen anfallen. Wegen diesen Ersparnissen lohnt sich der Einsatz von Open Source in Versicherungsunternehmen, auf deren branchenbedingte Besonderheiten im weiterführenden Text noch eingegangen wird. Weiterhin ist festzuhalten, dass Versicherungshäuser, welche Open Source einsetzen nicht in Gefahr laufen, in Herstellerabhängigkeit zu geraten.[41] Dies ist durch die Nutzung des offen Quellcodes und der Unterstützung durch die Anwendergemeinde begründet.[42] Ein Zwang zur Nutzung neuer Versionen wegen eines Supportablaufes kann nicht ausgeübt werden. Zudem besteht – wegen des einseh- und anpassbaren Codes – nicht die Erforderlichkeit, weitere Produkte des Herstellers zu verwenden.

[38] Vgl. Schaaf, A. (2013), S. 25
[39] Vgl. ebenda, S. 25
[40] Vgl. ebenda, S. 26
[41] Vgl. Königs, O. (2012), S. 19
[42] Vgl. Schaaf, A. (2013), S. 31 f.

Die Beachtung der genannten Punkte begründet, dass Versicherungen den Einsatz von Open Source forcieren und motiviert diese Ausarbeitung anzufertigen.

Historie von Open Source

Open Source-Software ist bereits lange mit der Geschichte der Informationstechnologie verzahnt. Die Ursprünge liegen in den 1960er Jahren, in denen das Geschäftsmodell im IT-Vertrieb darin bestand, Hardware zu verkaufen.[43] Software wurde als Nebenprodukt geliefert. Eine Änderung dieser Tatsache stellte sich ein, als die Systeme stets komplexer wurden.[44] Vor allem bei der Betrachtung des Preisverfalls der Hardware, welcher sich ab den 1970er einstellte, veranlasste Hersteller, die bislang frei verfügbare Software Lizenzen zu unterwerfen. Hierdurch konnten weiter Gewinne abgeschöpft werden.

Die Tendenz, Software kostenlos zur Verfügung zu stellen wurde ab 1983 vom US-amerikanischen Programmierer Richard Matthew Stallman geprägt. Das Anliegen von ihm und seiner Non-Profit-Organisation GNU war es, die Position von Software-Endverbrauchern zu stärken. Dies sollte vorwiegend durch die sogenannte „Free Software“ geschehen.[45] Die freie Software konnte sich im Laufe der anschließenden Jahre gegen proprietäre Softwarehäuser durchsetzen und etablieren.

Der Name Open Source entstand schließlich im Jahr 1998. Hauptanliegen war hierbei den Quellcode öffentlich einsehbar zu machen.[46] Dass es sich hierbei um eine erfolgreiche Marketingkampagne handelte ist vor allem daran ersichtlich, dass mehrere Hersteller – vor allem der damalige Internet-Browser-Hersteller Netscape Communications – unter großer Ankündigung in der internationalen Fachpresse den Source-Code für die selbst hergestellte Software freigab. Dass eine Notwendigkeit bestand, auch die Produkte der Open Source-Bewegung Lizenzbedingungen zu unterwerfen, wurde schnell deutlich.[47] Zwar bestanden Bestrebungen, die Lizenzen kostenlos zur Verfügung zu stellen, um eine Verbreitung der Software zu ermöglichen, die Kosten für die Verbreitung durch den Hersteller zu relativieren.[48] Auf eine weitergehende Betrachtung dieser Thematik auf aufgrund zeitlicher Vorgaben für diese Arbeit verzichtet.

In den letzten Jahrzehnten stellte die Wartung und Pflege für Open Source-Produkte eine Hauptherausforderung dar.[49] Dies geschieht vorwiegend durch die vorhandene Open Source-Community, welche im später folgenden Teil 7 bewertet werden wird. Die regelmäßi-

[43] Vgl. Saleck, T. (2005b), S. 11
[44] Vgl. ebenda, S. 11
[45] Vgl. Raggi, E./Thomas, K./van Vugt, S. (2011), S. 560
[46] Vgl. Böhnlein, I. (2003), S. 12
[47] Vgl. Saleck, T. (2005b), S. 11
[48] Vgl. ebenda, S. 9
[49] Vgl. Saleck, T. (2005a), S. 63

ge Wartung durch die Programmierer-Gemeinde verdeutlicht jedoch, dass auch in Zukunft die Betreuung von Open Source gewährleistet sein wird. Mit einer weiteren Etablierung und einem fortgesetzten Einsatz darf auch zukünftig gerechnet werden.

Besonderheiten im Versicherungsgewerbe

Jeder Wirtschaftszweig und jede Branche hat eigene Bedürfnisse bezüglich der IT-Infrastruktur.[50] Hieraus ergibt sich, dass eine variierende Nutzung von IT-Diensten abhängig von dem zu betrachtenden Industriezweig vorliegt. Als Beispiel darf die IT-Branche, und damit auch die Mehrzahl der im TecDAX notierten Unternehmen, genannt werden. Diese hat eine Vorreiterrolle bezüglich des Einsatzes von Informationstechnik eingenommen.[51] Neue Erkenntnisse und Produkte können hier eine schnelle Umsetzung finden.

Bereits in der Einleitung wurde beschrieben, dass diese Ausarbeitung die Verwendung von Open Source im Versicherungsbereich diskutiert. Aus diesem Grund ist es essentiell zu prüfen, welche Begebenheiten in der Versicherungsbranche existieren. Diese bilden eine Grundlage für weitere Überlegungen und dienen zudem als Bewertungshilfe für den ab Kapitel 7 dargelegten Kriterienkatalog.

Zunächst muss festgehalten werden, dass Versicherungsprodukte Erklärungsbedarf mit sich führen.[52] Kunden haben, anders als bei einer Vielzahl von Konsumgütern, ein entsprechendes Bedürfnis nach Erläuterung der angebotenen Versicherungspolicen. Dies ist vor allem deshalb von Bedeutung, da regelmäßig der Eindruck beim Verbraucher entsteht, dass die verschiedenen Angebote komplett miteinander substituierbar sind.[53] Darüber hinaus wird Versicherungsunternehmern und -maklern regelmäßig vorgeworfen, nicht im Sinne der potentiellen und bereits bestehenden Kunden zu handeln, sondern lediglich für ihre Provision zu arbeiten.[54] Diese Argumente führen dazu, dass in der Gesellschaft eine geringe Vertrauenswürdigkeit gegenüber Versicherungen besteht.[55] Als Konsequenz hieraus ist ein rücksichtsvolles und überlegtes Vorgehen bei allen strategischen und taktischen Entscheidungen notwendig. Eine Vorreiterrolle im Einsatz technologischer Neuerungen – wie im einleitenden Abschnitt dieses Kapitels für die IT-Branche aufgeführt – kann die Versicherungswirtschaft unter diesen Voraussetzungen nicht einnehmen. Der britische Finanzdienstleister Barclays PLC hat nach Recherchen der Daily Mail im vergangenen Jahr über 27.000 Datensätze an Kundendaten durch ein Sicherheitsleck verloren.[56] Ein vergleichbares Vorkommnis bei einem deutschen Versicherungshaus würde einen extremen Reputationsverlust

[50] Vgl. RES Software (o. J.)
[51] Vgl. Boes, A. u. a. (2012), S. 12
[52] Vgl. Mayerhofer, W. (2009), S. 2
[53] Vgl. ebenda, S. 1
[54] Vgl. ebenda, S. 57; vgl. dazu auch: Beenken, M. (2013), S. 155
[55] Vgl. Zwior, T. (2014)
[56] Vgl. Gallagher, I./Condron, S./Watkins, S. (2014)

nach sich ziehen. Empfehlenswert ist es für das Versicherungsgewerbe deshalb, Neuerungen der Informationstechnologie erst dann ein- und umzusetzen, wenn sie als etabliert bezeichnet werden dürfen. Dies ist auch die Tendenz, die von der Mehrheit der deutschen Versicherungsunternehmen verfolgt wird.[57]

Ein weiterer Grund für die deutschen Versicherer, neue Technologien erst nach deren Akzeptanz als verlässliches Instrument einzusetzen ist das konservative Verhalten der Branche. Diese Handlungsweise ist in erster Linie der Tatsache geschuldet, deutsche Versicherungsbetriebe überwiegend problemfrei durch die Finanzkrise von 2008 und 2009 danach zu bringen und stabil aufzutreten, was sich auch als erfolgreich bewährt hat.[58] Ein sich dadurch einstellender Nebeneffekt ist die behutsame Annahme von Neuerungen der Informationstechnologie. Ferner wird diese Tatsache dadurch bestärkt, dass das meist langjährig beschäftigte IT-Führungspersonal in Versicherungen gerne auf bewährte Systeme vertraut.[59]

Neben den ideologischen Faktoren spielen für Versicherungen auch praktische Themen eine Rolle, die einen vorerst gemäßigten Einsatz von neuer IT-Infrastruktur bestärkt. Dieser ist, dass in diesen Unternehmen häufig ältere informationstechnische Systeme zu finden sind.[60] Eine entsprechende Anbindung einer Neuerung innerhalb der IT scheitert entsprechend bereits an der Schnittstellenproblematik, welche sich erst im Laufe der Zeit lösen lässt.

Übertragen auf den Einsatz einer Cloud basierend auf dem Open Source-Prinzip bedeutet dies, dass eine Einführung grundsätzlich möglich ist. Dies ist unter anderem auch der Thematik geschuldet, dass Cloud Computing als etablierte Technologie betrachtet werden darf. Es empfiehlt sich für Versicherungen, dies jedoch zunächst in kleinen Schritten anzugehen, was sich auch im praktischen Teil dieser Ausarbeitung wiederfinden wird.

4 Private Clouds

Autoren: Amelie Böer, Thomas Enderle, Matthias Holzwarth, Dominik Schrahe

Nach der organisatorischen Unterteilung der Clouds ist es Ziel dieses Teils, die Art einer Cloud aus Kapitel 2.2 zu bestimmen, welche für Versicherungen am geeignetsten ist.

Die Vorteile der Public Cloud sind bereits im vorangegangenen erwähnt. Versicherungshäuser können hier speziell vom Auslagern der Daten profitieren. Zu erwähnen ist jedoch, dass diese positiven Aspekte nicht unbedingt Anwendung finden können, wenn die Systemzuver-

[57] Vgl. Ihlein, J. (2014)
[58] Vgl. RP Online (2010)
[59] Vgl. Ihlein, J. (2014)
[60] Vgl. ebenda

lässigkeit nicht ausreichend gewährleistet ist.[61] Zudem bestehen Anforderungen bezüglich des Datenschutzes, deren Einhaltung unabdingbar und gesetzlich vorgeschrieben ist.[62] Eine genaue Spezifikation dieses Themenbereiches darf – mit dem Bezug zur vorliegenden Praxis – dem anschließenden Unterkapitel entnommen werden.

Datenschutz und Datensicherheit

Datenschutz und Datensicherheit stellen eine unternehmerische Notwendigkeit dar.[63] Sie müssen besonders bei Versicherungen eine stetige Beachtung finden, da in diesem Geschäftsfeld besonders häufig mit personenbezogenen Daten gearbeitet wird. Es sollen die betroffenen Personen, die Versicherungsnehmer, gesetzlich geschützt werden.[64] Durch die Berichterstattung bezüglich des NSA-Skandals in den letzten eineinhalb Jahren verspüren diese eine berechtigte Verunsicherung.[65] Ein weiterer Grund, der darlegt, dass die Materie des Datenschutzes einen hohen Stellenwert in Öffentlichkeit und Politik hat, ist daran festzumachen, dass der deutsche Kommissar der Europäischen Union, Günther Oettinger, sein Amt mit der Zielsetzung antritt, die EU-Datenschutzreform voranzutreiben.

Der Schutz von der Daten der Kunden muss eine hohe Qualität haben, sodass eine Weitergabe an Dritte ausgeschlossen ist.[66] Ist kein ausreichender Schutz gewährleistet, kann es zu einer Preisgabe der Daten kommen. Hiermit gehen Schadenersatzforderung und -ansprüche der Kunden einher.[67] Weiterhin ist dieser Bereich – unabhängig von der Themenstellung – priorisiert zu behandeln, da bei einer ungewollten Veröffentlichung von Kundeninformationen mit einem Reputationsverlust zu rechnen ist. Dies rührt daher, da eine Verpflichtung besteht, in diesem Fall selbständig an die Öffentlichkeit zu treten.[68] Dies kann beispielsweise durch eine Veröffentlichung in zwei überregionalen Tageszeitungen geschehen.[69] Ein langjährig erarbeiteter Ruf wird dadurch beschädigt, sodass das Vertrauen der Kunden geschmälert wird. Zudem wirkt sich ein solcher Vorfall negativ auf das Neukundengeschäft aus. Eine juristische Sicherheit – um zu gewährleisten, dass Mitarbeiter sorgsam mit Kundendaten umgehen – kann durch Arbeitsvertrag und Betriebsvereinbarung gelöst werden. Organisatorische Sicherheit entsteht durch ein Risk-Management, Schulungen und Zertifizierungen.

[61] Vgl. Diefenbach, S./Brüning, K. T./Rickmann, H. (2013), S. 24
[62] Vgl. §1 Bundesdatenschutzgesetz
[63] Vgl. §1 Bundesdatenschutzgesetz
[64] Vgl. Schneider, J. (2012), S. 21 ff.; vgl. dazu auch §1 Bundesdatenschutzgesetz
[65] Vgl. Vollmer, A. (2014)
[66] Vgl. Schneider, J. (2012), S. 39; vgl. dazu auch: Hansen, M. (2012), S. 82; vgl. dazu auch §10 Bundesdatenschutzgesetz
[67] Vgl. Schneider, J. (2012), S. 39
[68] Vgl. Ade, U. (2014)
[69] Vgl. ebenda

Dies wird von diversen Dienstleistungsunternehmen angeboten.[70] Die technische Sicherheit wird gewährleistet durch verschiedene EDV-Funktionen.[71] Ein Beispiel dafür ist eine Firewall.

Entscheidung für die Private Cloud

Da der Datenschutz bei Versicherungen stets eine übergeordnete Rolle spielt, kommt für zukünftige Überlegungen des Cloud-Einsatzes in Versicherungen ausschließlich die Private Cloud in Betracht. Mit ihr können die bereits erwähnten Vorteile der Private Cloud (Dynamik, Effizienz und Flexibilität) erreicht werden. Dabei werden die Grundlagen des Datenschutzes eingehalten und es kann von den Vorteilen der Virtualisierung profitiert werden.

In den folgenden Kapiteln wird die Private Cloud daher als Lösung der Einrichtung einer Cloud in einem Versicherungshaus betrachtet.

Arten von Private Clouds

Eine abschließend zu beantwortende Fragestellung bezieht sich auf die Form der Private Cloud, bei der es zu klären, welche zu verwenden ist. Hierbei wird zwischen der Insourced Private Cloud und der Outsourced Private Cloud unterschieden.[72]

Die Fragestellung, die das Versicherungshaus hier zu treffen hat, ist, ob die Services – welche in einer Private Cloud abgebildet werden sollen – auf einem eigenen oder einem firmenfremden Server abgelegt werden sollen. Die gespeicherten Daten der Insourced Private Cloud werden im eigenen Haus, die der Outsourced Private Cloud bei einem Dienstleister abgelegt.[73]

Zwar bietet die Outsourced Private Cloud einen höheren Sicherheitsaspekt als die Public Cloud, es sind dennoch Überlegungen hinsichtlich des bereits dargelegten Datenschutzaspektes zu tätigen. Die Literatur weißt hierbei insbesondere die Problematik aus, dass sich eine Datenübernahme zu einem anderen Anbieter problematisch gestaltet[74], da hierbei vertrauliche Kundendaten nach außen gegeben werden. Kritisch ist hierbei, dass der Auftraggeber (das Versicherungshaus) nach §11 des Bundesdatenschutzgesetzes für die Einhaltung der gesetzlichen Vorgaben verantwortlich ist. Die Brisanz ist vor allem dann gegeben, wenn der Anbieter, bei dem die Cloud gehostet wird, seine Bedingungen bezüglich des Datenschutzes ändert. Es ist zudem ratsam, diese EDV-technische Umstellung schrittweise durchzuführen: Durch die Wahl einer Insourced Private Cloud entfällt zusätzlicher Arbeitsaufwand wie die Anbieterbewertung. Erst wenn sich der Einsatz der Private Cloud für ein

[70] Vgl. Kionga, C. (2014)
[71] Vgl. ebenda
[72] Vgl. Barton, T. (2014), S. 45 f.
[73] Vgl. ebenda, S. 45 f.
[74] Vgl. Maguire, J. (2013)

Versicherungshaus als rentabel darstellt, ist eine spätere Umstellung auf eine Outsourced Private Cloud, mit dem damit verbundenen Dauerschuldverhältnis, zu überlegen.[75]

Unter der Betrachtung dieser Punkte ist für das Versicherungswesen vorerst die Insourced Private Cloud zu verwenden, welche im weiteren Verlauf dieser Arbeit betrachtet wird.

5 Vorstellung ausgewählter Produkte

Autoren: Amelie Böer, Thomas Enderle, Matthias Holzwarth, Dominik Schrahe

Im Folgenden sollen verschiedene Cloud-Produkte vorgestellt werden. Der Fokus wird hierbei auf Private Clouds liegen. Da jedoch Amazon mit seinen Web Services als Infrastrukturanbieter im Bereich der Public Clouds hervorsticht und viele Private Clouds Schnittstellen zu Amazons Angeboten bieten, sollen auch diese zu Beginn dargestellt werden. Darüber hinaus sollen schwerpunktmäßig die Private Clouds Eucalyptus, CloudStack, OpenStack und ownCloud betrachtet werden. Die Auswahl basiert einerseits auf den laut opencource.com wichtigsten Open Source-Projekten in 2014. Unter die Top-10 kamen sowohl OpenStack als auch ownCloud.[76] Da ersteres auf CloudStack basiert wurde auch dieses Produkt betrachtet. Eine große Bedeutung auf dem Private Cloud-Markt kommt auch Eucalyptus zu, weshalb dieses Produkt ebenfalls vorgestellt werden soll.

In einem weiteren Abschnitt werden auch einige PaaS-Implementierungen vorgestellt.

Amazon Web Services

Unter dem Begriff der Amazon Web Services (AWS) sind alle Cloud-Dienstleistungen Amazons zusammengefasst. Hierunter fallen diverse Dienste, angefangen von der Elastic Compute Cloud (EC2), durch die virtuelle Server bereitgestellt werden, über Messaging- und Datenbanksysteme bis hin zu Speicherdiensten.

Amazons Simple Storage Service (S3) stellt einen solchen Speicherdienst dar, wobei Daten in sogenannte Buckets vorliegen. Diese bekommen als Objekte Nummern zur Identifikation zugewiesen. Ein Zugriff kann unter anderem über RESTful[77] als offene Schnittstelle erfolgen.[78] Darüber hinaus wird mit Elastic Block Store (EBS) ein weiterer Speicherdienst zur persistenten Vorhaltung von Daten angeboten. Hiermit können sogenannte Volumes zur Speicherung von Daten und EC2-Instanzen angelegt werden, die mit einem frei wählbaren

[75] Vgl. Kionga, C. (2014)
[76] Vgl. Huger, J. W. (2014)
[77] REST steht für Representational State Transfer und stellt ein Web-Programmierparadigma dar. Dieses verlangt unter anderem, dass eine Webadresse genau zu einem Seiteninhalt führt und dieser auch zu einem späteren Zeitpunkt unter derselben Adresse erreichbar ist.
[78] Vgl. Barton, T. (2014), S. 63 ff.

Dateisystem zu formatieren sind. Die Volumes können im Betrieb jeweils nur in einer Instanz eingehängt werden.[79]

Eucalyptus

Der Name Eucalyptus steht für „Elastic Utility Computing Architecture for Linking Your Programs to Useful Systems“. Dabei handelt es sich um eine IaaS-Lösung, die zur Virtualisierung auf die Hypervisoren Xen und KVM oder vSphere von Vmware aufsetzt. Eucalyptus besteht aus den drei grundsätzlichen Komponenten Cloud Controller (CLC), Cluster Controller (CC) sowie Node Controller (NC). Auf jedem Knoten[80] wird ein Node Controller ausgeführt, der die Ressourcenauslastung des Knotens im Bezug auf Prozessoren, RAM und Festspeicher an den Cluster Controller sendet. Dieser verteilt die Last über alle Nodes im Cluster und meldet den Ressourcenzustand des gesamten Clusters an den Cloud Controller. Letzterer dient in jeder Eucalyptus-Installation als zentraler Zugriffspunkt sowohl für Benutzer als auch Administratoren. Dies gilt auch für Installationen geringer Größe, bei denen CLC und CC auf einem physikalischen Rechner laufen.

Eucalyptus legt besonderen Wert auf die Kompatibilität zu den Amazon Web Services. So ist das Modul zur Speicherung von Web-Objekten Walrus Kompatibel zur API von Amazons Simple Storage Service. Der interne Storage Controller benutzt dieselbe API wie Amazons Elastic Block Storage. Die Speicherdienste können auf beliebige Server im Cluster verteilt werden, bei kleinen Installationen ist auch eine Speicherung auf dem Cloud Controller möglich.

Neben den Speicherdiensten stellt Eucalyptus auch das Virtual Distribution Ethernet (VDE) genannte Netzwerkmodul bereit. Dieses sorgt dafür, dass Traffic zwischen den Knoten über einen virtuellen Switch und virtuelle Kabel läuft, sodass ein Subnetz entsteht.[81]

CloudStack

Bei CloudStack handelt es sich, wie bei Eucalyptus, um eine IaaS-Lösung, die Xen, KVM und vSphere unterstützt. Dabei besteht CloudStack aus einem Management-Server und Rechenknoten. Ersterer bietet eine Weboberfläche zur Administration sowie zum produktiven Zugriff und verteilt gleichzeitig die Instanzen je nach Ressourcenauslastung auf die Rechenknoten. Diese übernehmen schließlich die eigentlichen Berechnungen. Im Gegensatz zu anderen, in diesem Kapitel vorgestellten Private Cloud-Produkten, steht CloudStack nur in

[79] Vgl. Baun, C. u. a. (2011), S. 54 f.
[80] Als Node (dt. auch Knoten) bezeichnet man einzelne Rechner in einem Rechnerverbund. Jeder Knoten übernimmt bestimmte Aufgaben, wobei auch mehrere Knoten im Rechnerverbund aus Gründen der Ausfallsicherheit oder zur Erhöhung der Gesamtperformance zusammenarbeiten können.
[81] Vgl. Baun, C. u. a. (2011), S. 94 ff.

einer Community Version unter Open Source-Lizenz zur Verfügung. Darüber hinaus existieren mit der Enterprise- und der Service Provider Edition weitere Versionen.[82]

OpenStack

Die IaaS-Plattform OpenStack entstand in einer Zusammenarbeit von Rackspacke mit der NASA, wobei inzwischen diverse Unternehmen aus der Kommunikations- und IT-Branche die Entwicklung vorantreiben. Dabei stellt OpenStack kein fertiges Produkt dar, sondern besteht aus diversen Komponenten, durch die der hohe Funktionsumfang und die universelle Einsetzbarkeit zustande kommt. Die zahlreichen OpenStack-Entwickler sorgen in halbjährlichen Releasezyklen für neue Versionen.[83]

Darüber hinaus existieren noch diverse weitere Module, die teilweise für Kompatibilität zu Public Cloud-Anbietern wie Amazon sorgen. In diesem Bereich ist Swift als zu Amazons Simple Storage Service kompatiblem Objektspeicher zu nennen. Der Speicherdienst Cinder ist dagegen zu Amazons Elastic Block Storage kompatibel. Neben diesen Modulen wird mit Nova die Möglichkeit zur Bereitstellung virtueller Server gegeben. Falls ein detaillierteres Ressourcenmonitoring erforderlich wird, steht das Modul Ceilometer unter anderem zur Messung von CPU- und RAM- Auslastung zur Verfügung. Neutron ermöglicht es, eine virtuelle Netzwerkinfrastruktur bereitzustellen, wodurch einerseits Sicherheitsmechanismen wie Firewalls mit benutzerdefinierten Regelungen sowie Intrusion Detection Systeme bereitgestellt werden können. Darüber hinaus ist auch ein load balacing zwischen mehreren virtuellen Maschinen oder die Anbindung entfernter Nutzer per VPN möglich.[84] Das Zusammenspiel dieser und weiterer Komponenten in OpenStack wird in Abb. 3 dargestellt.

[82] Vgl. Baun, C. u. a. (2011), S. 102
[83] Vgl. Seidel, U. (2014), S. 14 f.
[84] Vgl. Yeluri, R./Castro-Leon, E. (2014), S. 135 f.

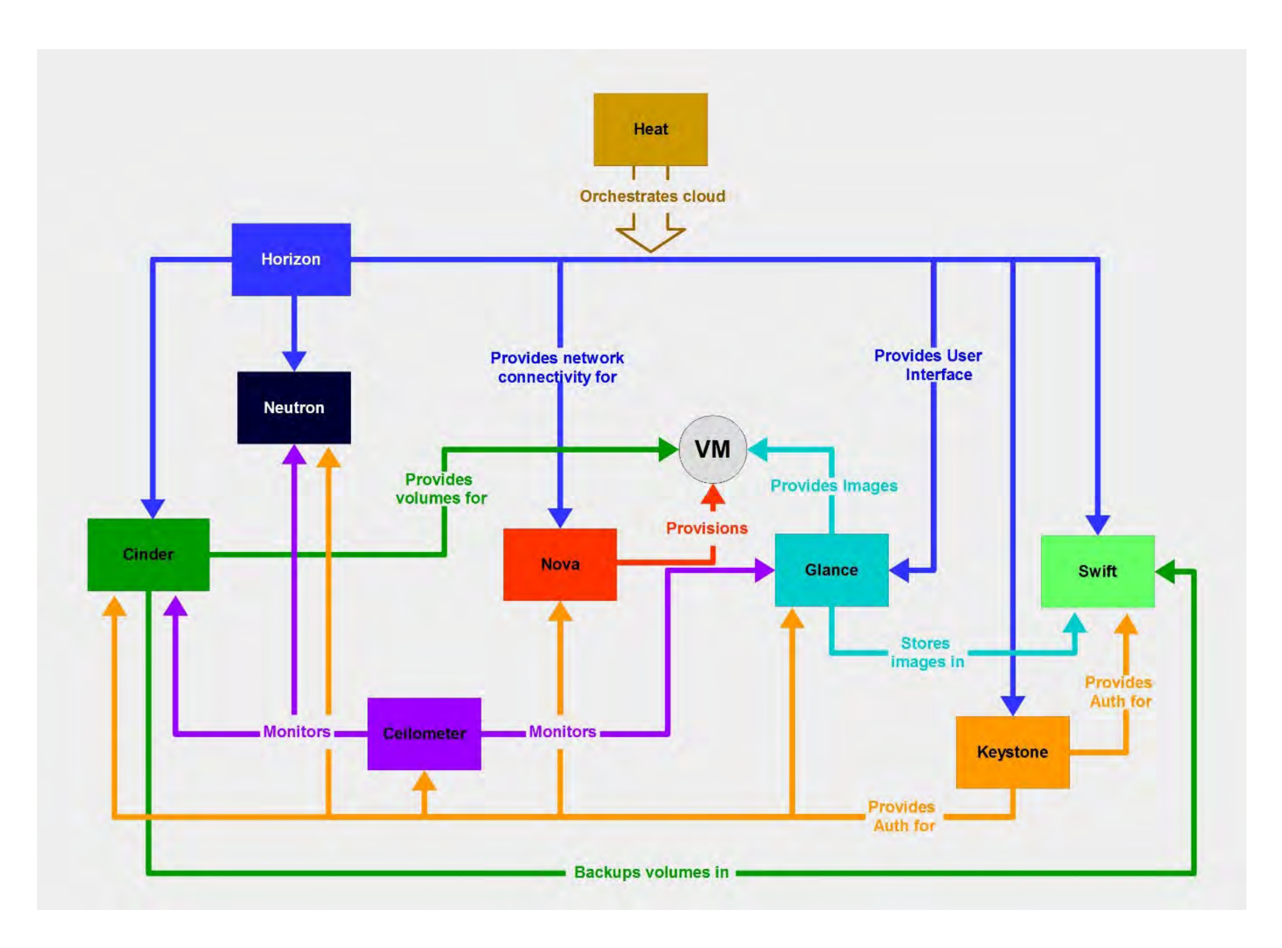


Abb. 3: Komponenten und ihre Zusammenarbeit in OpenStack[85]

ownCloud

ownCloud wird, wie die bereits vorgestellten Produkte, häufig in den Bereich IaaS eingeordnet. Diese Einordnung gilt insbesondere bei der vorwiegenden Betrachtung des bereitgestellten Speicherdienstes. Sofern die weiteren Funktionen zur Kalender- und Kontakteverwaltung betrachtet werden, ist eine Zuordnung in den Bereich SaaS möglich. Dennoch unterscheidet sich OwnCloud erheblich von den bereits vorgestellten Produkten. So wird ein Server in der Regel auf einer LAMP-Installation, bestehend aus Linux-Betriebssystem, Webserver, relationalem Datenbankmanagementsystem und PHP installiert. Die Verwendung eines Hypervisors ist nicht notwendig.[86] Die Community Edition von ownCloud enthält in seiner Grundfunktionalität Möglichkeiten zur Datei- Kalender- und Kontaktesynchronisation über das Internet. Auf die nicht unter Open Source Lizenz stehende Enterprise Edition wird vergleichend in Abschnitt 9.6 eingegangen. Die Dateiübertragung und Speicherung kann verschlüsselt erfolgen, wobei zusätzlich Funktionen wie ein Dateiversionsverlauf angeboten werden. Dateien können mit anderen Benutzen und Gruppen oder öffentlich per Link geteilt werden. Zur Benutzerverwaltung kann ein LDAP-Backend verwendet werden. Der Zugriff auf sämtliche Funktionen und die Administration ist per Weboberfläche möglich. Diese ermöglicht auch die

[85] Mit Änderungen entnommen aus: Seidel, U. (2014), S. 15
[86] Vgl. ownCloud (o. J.c)

direkte Anzeige und Bearbeitung von Text- sowie OpenDocument-Dateien. PDF und Bilddateien können ebenso angezeigt werden.[87] Daneben existieren Clients für Windows, Mac OS und Linux sowie Android und iOS zur Datensynchronisation. Mehrere ownClouds können durch eine Server-to-Server Sharing genannte Funktion verbunden werden.[88]

Der Zugriff auf Daten ist zusätzlich per WebDAV möglich, wodurch die Anbindung alle Plattformen vereinfacht wird. Auch die Synchronisation von Kalendern und Kontakten kann per CalDAV bzw. CardDAV erfolgen, was die Interoperabilität mit den meisten Plattformen und Programmen ermöglicht. Für zusätzliche Funktionen stehen darüber hinaus diverse Apps bereit.[89] Neben der kostenlosen, unter Open Source-Lizenz stehenden, Version von ownCloud existiert eine Enterprise Edition mit speziellen Funktionen für Unternehmen.[90]

Weitere IaaS-Implementierungen

Bei Open Nebula handelt es sich ebenfalls um eine IaaS-Lösung, die Xen, KVM oder vSphere als Hypervisor nutzt. Open Nebula ist in der Lage, einzelne Instanzen im laufenden Betrieb auf andere Knoten zu verschieben und besitzt als einziges vorgestelltes Produkt die Fähigkeit zum High Performance Computing. Für diese HPCaaS (High Performance Computing as a Service) genannte Funktion werden mehrere Knoten gruppiert, sodass ihre Rechenleistung gemeinsam zur Verfügung steht. Im Gegensatz zu Eucalyptus und OpenStack besteht bei Open Nebula keine Kompatibilität zu den Speicherdiensten der Amazon Web Services.

Die IaaS-Lösung Nimbus benutzt Xen und KVM als Hypervisor. Der integrierte Speicherdienst Cumulus ist in der Lage auf Amazons Simple Storage Service zurückzugreifen.[91]

Platform as a Service

Im Bereich PaaS ist die Auswahl deutlich kleiner wobei die Private Cloud-Annieter größtenteils versuchen, die von der Google App Engine vorgegebene Funktionalität bestmöglich nachbilden zu können. Googles App Engine stellt dabei eine Public Cloud Plattform zur Entwicklung von Webanwendungen bereit. Softwareentwickler bekommen folglich eine Programmier- sowie Ausführungsumgebung zur Verfügung gestellt, wobei die zuverlässige Bereitstellung und Administration der Server durch Google übernommen wird. Für die Anwendungsentwicklung stehen Java und Python bereit. Darüber hinaus stellt Google weitere Dienste unter anderem zur Datenspeicherung oder zum performanten Caching bereit.

[87] Vgl. The ownCloud developers (2014), S. 7 ff.
[88] Vgl. The ownCloud developers (2014), S. 23 f.
[89] Vgl. ebenda, S. 15 ff.
[90] Vgl. ownCloud (o. J.f)
[91] Vgl. Baun, C. u. a. (2011), S. 99 f.

Das Ziel die genannte Funktionalität auch im Rahmen einer Private Cloud zur Verfügung stellen zu können, verfolgt das Open-Source-Projekt AppScale. Auch auf dieser Plattform ist die Anwendungsentwicklung auf Basis von Java und Python möglich, wobei auch die bereits genannten Google-typischen Infrastrukturdienste bereitgestellt werden. AppScale kann einerseits auf einer bestehenden IaaS-Plattform wie Eucalyptus aufbauen, genauso ist der unmittelbare Betrieb auf einem Xen Hypervisor möglich.

Neben AppScale existiert mit typhoonAE eine weitere freie PaaS, die der Google App Engine recht nahe kommt. Zwar wird hier nur die Entwicklung von Python-Anwendungen unterstützt, die von Google bereitgestellten Infrastrukturdienste werden jedoch auch von typhoonAE emuliert. Ein zentraler Unterschied zu AppScale stellt der unkomplizierte Betrieb direkt auf Linux-Plattformen dar. Somit wird keine IaaS-Lösung und kein Hypervisor als Basis benötigt.

6 Sicherheit

Autoren: Amelie Böer, Thomas Enderle, Matthias Holzwarth, Dominik Schrahe

In diesem Kapitel werden drei Sicherheitskonzepte bezüglich des Datenschutzes vorgestellt, die für Cloud-Anwendungen relevant sein können. Das ist zum einen die Zugriffskontrolle am Beispiel der (Benutzer-)Authentifizierung, zum anderen die Datenintegrität und zum letzten die Verschlüsselung. Sie werden in diesem Kapitel vorgestellt und erläutert. Dabei wird auf verschiedene Möglichkeiten der Umsetzung eingegangen.

6.1 Authentifizierung

Nach dem Duden bedeutet „authentifizieren“ so viel wie beglaubigen oder die Echtheit von etwas bezeugen.[92] In der IT ist damit meist die Beglaubigung der Identität einer Person gemeint. Grundsätzlich kann die Authentifizierung über drei Wege geschehen. Die Möglichkeiten sind:

- durch **Wissen**,
  beispielsweise durch ein Passwort
- durch **Besitz**,
  über sogenannte Tokens
- durch **biometrische Merkmale**,
  zum Beispiel mittels Fingerabdruck[93]

[92] Vgl. o. V. (2013)
[93] Vgl. Tsolkas, A./Schmidt, K. (2010), S. 127

In dieser Arbeit werden nur die ersten beiden Wege betrachtet. Als Beispiel für wissensbasierte Authentifizierung wird Benutzername und Passwort beschrieben, für besitzbasierte Authentifizierung werden Tokens vorgestellt. Anschließend werden beide zusammen zur sogenannten Zwei-Faktor-Authentifizierung (2FA) zusammengeführt.

Benutzername und Passwort

Die Kombination aus Benutzername und Passwort ist die verbreitetste Form der Authentifizierung.[94] Sie besteht aus einer öffentlichen (Benutzername) und einer privaten Information (Passwort).

Der Benutzer gibt seinen Benutzernamen und sein Passwort auf einer Anmeldemaske ein. Anschließend wird überprüft, ob der Benutzername vorhanden ist und ob das Passwort zum Benutzernamen passt.

Wird das Passwort stets geheim gehalten, so stellt die Kombination aus Benutzername und Passwort in der Regel eine ausreichend sichere Authentifizierungsoption dar.[95]

Token

Ein Token ist eine Hardware, die Passwörter, PINs, TANs oder ähnliches generieren kann. Der Benutzer kann den generierten Code dann über ein kleines Display ablesen und anschließend in einer Anmeldemaske eingeben. Ein generierter Code kann immer nur einmal zur Authentifizierung genutzt werden, danach verfällt er.[96]

Zwei-Faktor-Authentifizierung

Möchte ein Unternehmen den Benutzern seiner IT-Dienste verstärkte Authentifizierung ermöglichen, so kann eine 2FA eingeführt werden. Sie basiert darauf, dass der Benutzer zur Authentifizierung Wissen und Besitz nachweisen muss. Dazu werden ein Passwort und ein Token-Code abgefragt. Dies führt zu einer höheren Sicherheit, da nur das Wissen oder nur der Besitz nicht zu einer erfolgreichen Authentifizierung führen.[97]

6.2 Verschlüsselung

Die Verschlüsselung hat das Ziel einen **Klartext** so über einen abhörbaren Kanal zu übertragen, dass ein unberechtigter Dritter den Klartext nicht erhält. Dazu wird der Klartext **verschlüsselt**. Der Klartext wird so verfremdet, dass er ohne zusätzliche Informationen nicht wieder in seinen ursprünglichen Zustand zu überführen ist. Die Information, die zur Ent-

[94] Vgl. Balzert, H. (2011), S. 154
[95] Vgl. Tsolkas, A./Schmidt, K. (2010), S. 128
[96] Vgl. ebenda, S. 141 f.
[97] Vgl. Tsolkas, A./Schmidt, K. (2010), S. 143 f.

schlüsselung gebraucht wird, wird in dieser Arbeit als **Key** bezeichnet. Das Überführen des verschlüsselten Textes in den Klartext wird als **Entschlüsselung** bezeichnet.

Zur Verschlüsselung von Daten existieren zwei grundlegende Techniken. Zum einen die **symmetrische** und zum anderen die **asymmetrische Verschlüsselung**. Die symmetrische Verschlüsselung wurde bereits von Caesar zur militärischen Kommunikation genutzt.[98] Im Gegensatz dazu ist die asymmetrische Verschlüsselung ein relativ neues Konzept, dass in den 70er Jahren des 20. Jahrhunderts entwickelt wurde.[99] Beide Techniken werden im Folgenden erläutert.

symmetrische Verschlüsselung

Bei der symmetrischen Verschlüsselung wird zur Ver- und Entschlüsselung derselbe Key verwendet. Nachteil der symmetrischen Verschlüsselung ist, dass der Key vorher auf einem sicheren Weg übertragen werden muss.[100] Abb. 4 zeigt das Prinzip der symmetrischen Verschlüsselung grafisch.

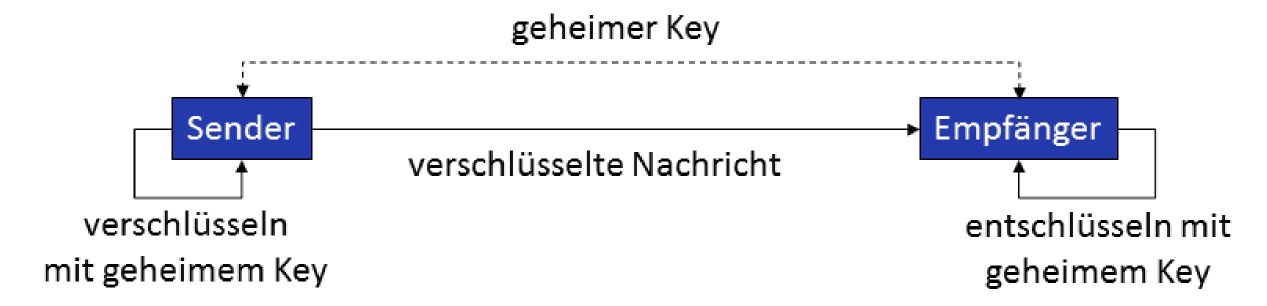


Abb. 4: Prinzip der symmetrischen Verschlüsselung

Einer der bekanntesten symmetrischen Verschlüsselungsalgorithmen ist der Advanced Encryption Standard (AES). AES ist eine Variante des Rijndael-Algorithmus[101] und darf von US-amerikanischen Behörden zur Verschlüsselung von geheimen Informationen genutzt werden. Das NIST überprüft den AES-Standard alle fünf Jahre[102] und somit kann davon ausgegangen werden, dass AES sicher ist.[103]

Bei AES handelt es sich um eine Blockverschlüsselung, die eine Blocklänge von 128 Bit (16 Byte) nutzt. Als Schlüssellängen stehen 128, 192 oder 256 Bit (16, 24 oder 32 Byte) zur Verfügung. Aus diesem Grund werden die drei verschiedenen Möglichkeiten AES-128, AES-192 und AES-256 genannt.[104]

[98] Vgl. Küsters, R./Wilke, T. (2011), S. 7
[99] Vgl. ebenda, S. 8
[100] Vgl. ebenda, S. 7
[101] Vgl. National Institute of Standards and Technology (2001), S. 5
[102] Vgl. ebenda, S. i
[103] Vgl. Küsters, R./Wilke, T. (2011), S. 71
[104] Vgl. National Institute of Standards and Technology (2001), S. 5

Beim Einsatz einer Blockverschlüsselung wird die zu verschlüsselnde Nachricht in Blöcke fester Länge unterteilt wird. Die Blöcke werden anschließend einzeln verschlüsselt. Ist die Nachricht kein ganzzahliges Vielfaches der gewählten Blocklänge, so wird der letzte Block mit weiteren Zeichen aufgefüllt.[105]

Bei AES werden, abhängig von der Schlüssellänge, 10 (AES-128), 12 (AES-192) oder 14 Runden (AES-256) des nachfolgend beschriebenen Algorithmus durchlaufen. Der Algorithmus arbeitet intern auf einem 4 x 4 Byte-Array, welches auch State Array genannt wird.[106]

Zu Beginn wird einmal die AddRoundKey()-Funktion ausgeführt, danach besteht jede Runde aus vier Schritten, die durch folgende Funktionen repräsentiert werden. Einzige Ausnahme ist die letzte Runde, bei der MixColumns() nicht durchgeführt wird:[107]

1. SubBytes()
   Hier werden die Werte des State Arrays mit Hilfe einer Substitutionsbox ersetzt.[108]

2. ShiftRows()
   Hier werden die Zeilen des State Arrays um einen bestimmten Offset verschoben.[109]

3. MixColumns()
   Hier werden die Elemente des State Arrays innerhalb einer Spalte vermischt.[110]

4. AddRoundKey()
   Hier werden die Elemente einer Spalte des State Arrays mit einem Rundenschlüssel XOR-verknüpft.[111]

1976 entwickelten Whitfield Diffie und Martin E. Hellman ein Verfahren[112], wie sich zwei Kommunikationspartner über einen abhörbaren Kanal auf einen geheimen Key einigen können, ohne dass dieser vollständig übertragen wird.[113] Mit diesem Verfahren, dem Diffie-Hellman-Schlüsselaustausch (DH), kann das Problem umgangen werden, den geheimen Key auf einem sicheren Weg übertragen zu müssen.[114]

Die Sicherheit des DH-Schlüsselaustausches basiert auf der Annahme, dass diskrete Logarithmen schwer zu berechnen sind.[115] Es existieren aber verschiedene Verfahren, mit denen diskrete Logarithmen relativ einfach zu berechnen sind. Aus diesem Grund wurde eine Vari-

[105] Vgl. Küsters, R./Wilke, T. (2011), S. 37
[106] Vgl. National Institute of Standards and Technology (2001), S. 9
[107] Vgl. ebenda, S. 14
[108] Vgl. ebenda, S. 15 f.
[109] Vgl. ebenda, S. 17
[110] Vgl. ebenda, S. 17 f.
[111] Vgl. National Institute of Standards and Technology (2001), S. 18 f.
[112] Vgl. Küsters, R./Wilke, T. (2011), S. 2
[113] Vgl. Bartosch, M. (2008)
[114] Vgl. Küsters, R./Wilke, T. (2011), S. 2
[115] Vgl. ebenda, S. 170

ante des DH-Schlüsselaustausches mit elliptischen Kurven entwickelt, für die solche Verfahren nicht bekannt sind.[116] Der DH-Schlüsselaustausch mit elliptischen Kurven wird ECDH abgekürzt.

(EC)DH nutzt einen ausgehandelten Key während der gesamten Sitzung. Das bereitet Probleme, wenn der geheime Key von einem Angreifer erraten oder berechnet wird. Dann hat er die Möglichkeit die gesamte Kommunikation zu entschlüsseln. Deshalb existiert eine Variante, die (Elliptic Curve) Diffie-Hellman Ephemeral, abgekürzt (EC)DHE, heißt. Hier wird in regelmäßigen Abständen[117], beispielweise bei jeder Verbindung, ein neuer Key ausgehandelt. So wird Perfect Forward Secrecy (PFS) für (EC)DHE ermöglicht.[118] PFS schützt davor, dass die übertragenen Daten im Nachhinein, entschlüsselt werden können.[119] Das liegt daran, dass der geheime Key nach jeder Verbindung gelöscht und nicht wiederverwendet wird.[120]

Wichtig ist, dass die Daten des Schlüsselaustausches digital signiert sind, da der DH-Schlüsselaustausch sonst anfällig für Man-in-the-Middle-Angriffe (MITM) ist.[121] Abb. 5 zeigt schematisch einen MITM-Angriff.

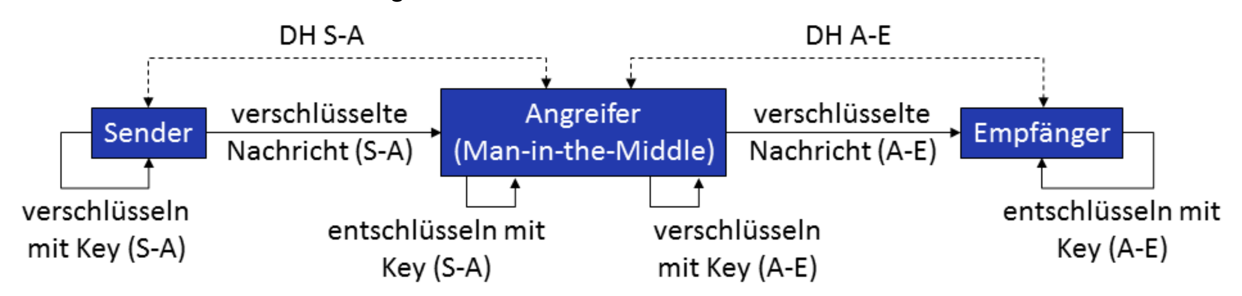


Abb. 5: Man-in-the-Middle-Angriff

Beim MITM-Angriffsszenario auf DH werden zwei Schlüsselaustäusche durchgeführt. Einmal zwischen dem Sender und dem Angreifer, und einmal zwischen dem Angreifer und dem Empfänger. Somit kann der MITM die gesamte Kommunikation abhören.[122]

Aufgrund des beschriebenen Szenarios ist es wichtig, dass die Integrität des Schlüsselaustausches sichergestellt wird. Dazu können die im Abschnitt Integrität beschriebenen Techniken eingesetzt werden.

[116] Vgl. ebenda, S. 172
[117] Vgl. o. V. (o. J.b)
[118] Vgl. Bakker, P. (2013)
[119] Vgl. Schmidt, J. (2013)
[120] Vgl. Thome, N./Haynberg, R. (2014)
[121] Vgl. Bartosch, M. (2008)
[122] Vgl. o. V. (o. J.b)

asymmetrische Verschlüsselung

Die zweite Möglichkeit eine Verschlüsselung zu realisieren ist die asymmetrische Verschlüsselung. Hierbei werden zum ver- und entschlüsseln unterschiedliche Keys genutzt. Jeder Kommunikationspartner besitzt einen Public Key, der zum Verschlüsseln dient. Der **Public Key** kann und soll veröffentlicht werden. Im Gegensatz dazu muss der **Private Key** geheim gehalten werden. Mit ihm werden die Nachrichten wieder entschlüsselt.[123] Auch mit diesem Ansatz wird ein abhörsicheres Übertragen eines geheimen Keys umgangen.[124] Der Nachteil hierbei ist wiederum, dass asymmetrische Verschlüsselungsverfahren in der Regel langsamer als symmetrische arbeiten.[125] Das Prinzip der asymmetrischen Verschlüsselung zeigt Abb. 6.

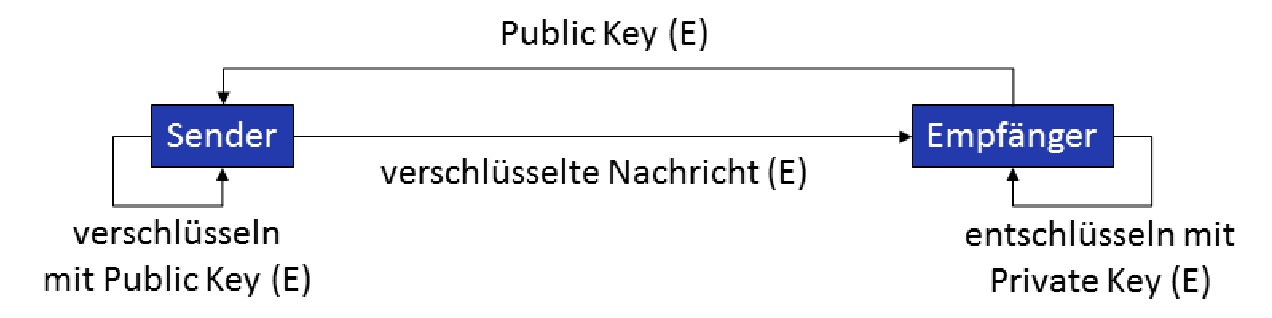


Abb. 6: Prinzip der asymmetrischen Verschlüsselung

Ein sehr weit verbreitetes Verfahren zur asymmetrischen Verschlüsselung ist das Rivest, Shamir und Adleman-Verfahren (RSA).[126] Es basiert, wie alle asymmetrischen Verfahren, auf einer Einwegfunktion. Eine Einwegfunktion ist eine Funktion, deren Funktionswert einfach zu berechnen ist. Die Umkehrfunktion, also die Berechnung eines Arguments, ist aber nur mit viel Aufwand zu berechnen. Das bedeutet, dass das Finden eines Arguments zwar theoretisch möglich, bei ausreichender Größe des Arguments aber praktisch nicht möglich ist.

Die Einwegfunktion, die bei RSA eingesetzt wird, ist die Multiplikation von Primzahlen. Das Produkt zweier Primzahlen ist sehr einfach zu berechnen. Die sogenannte Faktorisierung, also die Zerlegung eines Produkts in seine Primfaktoren, ist hingegen nur schwer durchzuführen.[127]

Aufgrund dieser Tatsachen ist es kaum möglich aus dem Public Key den Private Key zu berechnen. Sollte die verfügbare Rechnerkapazität in Zukunft ausreichen, um den Private

[123] Vgl. Witt, K.-U. (2014), S. 165 f.
[124] Vgl. Küsters, R./Wilke, T. (2011), S. 8
[125] Vgl. ebenda, S. 175
[126] Vgl. Witt, K.-U. (2014), S. 168
[127] Vgl. ebenda, S. 166 f.

Key relativ schnell zu berechnen, so können einfach größere Keypaare generiert werden.[128] Allerdings ist zu beachten, dass für Quantencomputer effiziente Algorithmen zur Faktorisierung existieren. Es ist jedoch zweifelhaft ob jemals Quantencomputer mit genügend Kapazität zur Verfügung stehen werden.[129]

hybride Verschlüsselung

Aufgrund der Nachteile von symmetrischer und asymmetrischer Verschlüsselung, wird heutzutage oft eine Kombination der beiden Verfahren eingesetzt. Die **hybride Verschlüsselung** kombiniert symmetrische und asymmetrische Verschlüsselung so, dass man ein effizientes und trotzdem sicheres Verfahren erhält. Die Funktionsweise einer hybriden Verschlüsselung ist in Abb. 7 dargestellt.



Abb. 7: Prinzip der hybriden Verschlüsselung

Der Sender generiert einen geheimen Key, der asymmetrisch mit dem Public Key des Empfängers verschlüsselt wird. Die eigentliche Nachricht wird symmetrisch mit dem geheimen Key verschlüsselt. Anschließend werden der verschlüsselte geheime Key und die verschlüsselte Nachricht an den Empfänger gesendet. Dieser kann den verschlüsselten geheimen Key mit seinem Private Key und anschließend die verschlüsselte Nachricht mit dem geheimen Key entschlüsseln.[130]

6.3 Integrität

Das Bundesamt für Sicherheit in der Informationstechnik definiert Integrität als „die Sicherstellung der Korrektheit (Unversehrtheit) von Daten […]".[131]

Die Integrität von Daten hat in diesem Kontext aber auch zusätzliche Bedeutungen, nämlich dass der Absender einer Nachricht nicht abstreiten kann, dass er der Autor ist (Verbindlich-

[128] Vgl. ebenda, S. 173
[129] Vgl. Böck, H. (2012)
[130] Vgl. Küsters, R./Wilke, T. (2011), S. 175
[131] Bundesamt für Sicherheit in der Informationstechnik (2009)

keit), sowie dass er nicht abstreiten kann, dass eine Nachricht von ihm unterzeichnet wurde (Authentizität). Alle drei Aspekte können über digitale Signaturen erreicht werden.[132]

Eine Möglichkeit die Integrität einer Nachricht sicherzustellen, bietet das RSA-Verfahren, welches im letzten Abschnitt vorgestellt wurde. Dafür wird das Verfahren „umgedreht“.[133] Dies ist schematisch in Abb. 8 dargestellt.

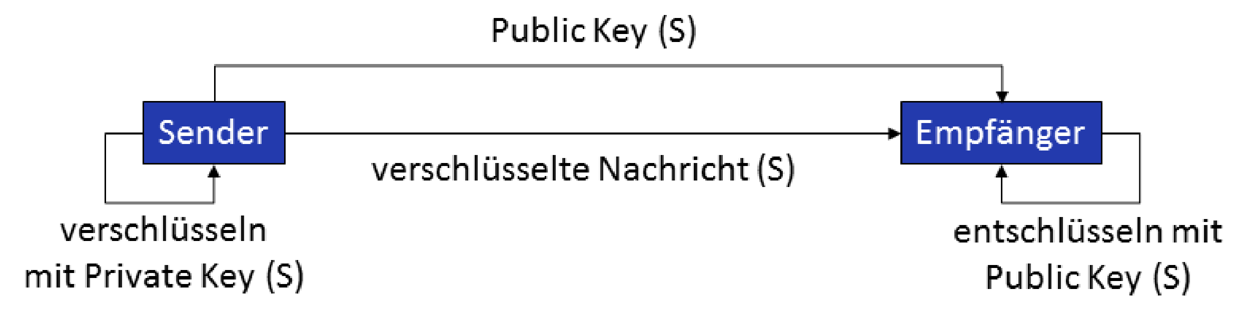


Abb. 8: Prinzip einer digitalen Signatur

Der Sender verschlüsselt seine Nachricht mit seinem Private Key. Der Empfänger kann die Nachricht nun mit dem Public Key des Senders entschlüsseln. Ist die eigentliche Nachricht identisch mit der signierten Nachricht, so kann der Empfänger davon ausgehen, dass tatsächlich der Sender die Nachricht geschickt hat und die Nachricht nicht manipuliert wurde.[134]

Problematisch bei dieser Vorgehensweise ist, dass zum einen die verschlüsselte Nachricht und die signierte Nachricht übertragen werden müssen. Da beide gleich groß sind, verdoppelt sich das übertragene Datenvolumen. Aus diesem Grund werden Hash-Funktionen eingesetzt, die zu einer Zeichenkette beliebiger Länge einen Hashwert einer bestimmten Länger berechnen. Hash-Funktionen sind Einwegfunktionen und kollisionssicher. Kollisionssicherheit bedeutet, dass zwei Eingangs-Zeichenketten niemals den gleichen Hash-Wert erhalten können.[135]

Ein Beispiel für eine solche Hash-Funktion ist Secure Hash Algorithm Version 2 (SHA2). SHA2 existiert in drei Ausführungen, die unterschiedlich lange Hash-Werte erzeugen. Die Benennung erfolgte, ähnlich wie bei AES, nach der Länge des Hash-Werts. SHA-256 erzeugt 256 Bit (32 Byte), SHA-384 erzeugt 384 Bit (48 Byte) und SHA-512 erzeugt 512 Bit (64 Byte) Lange Hash-Werte.[136]

Abb. 9 zeigt nun die verschlüsselte Kommunikation mit einem asymmetrischen Verfahren in Kombination mit Integritätsschutz über digitale Signaturen.

[132] Vgl. Witt, K.-U. (2014), S. 177
[133] Vgl. Witt, K.-U. (2014), S. 177
[134] Vgl. ebenda, S. 177 f.
[135] Vgl. Witt, K.-U. (2014), S. 178
[136] Vgl. National Institute of Standards and Technology (o. J.)

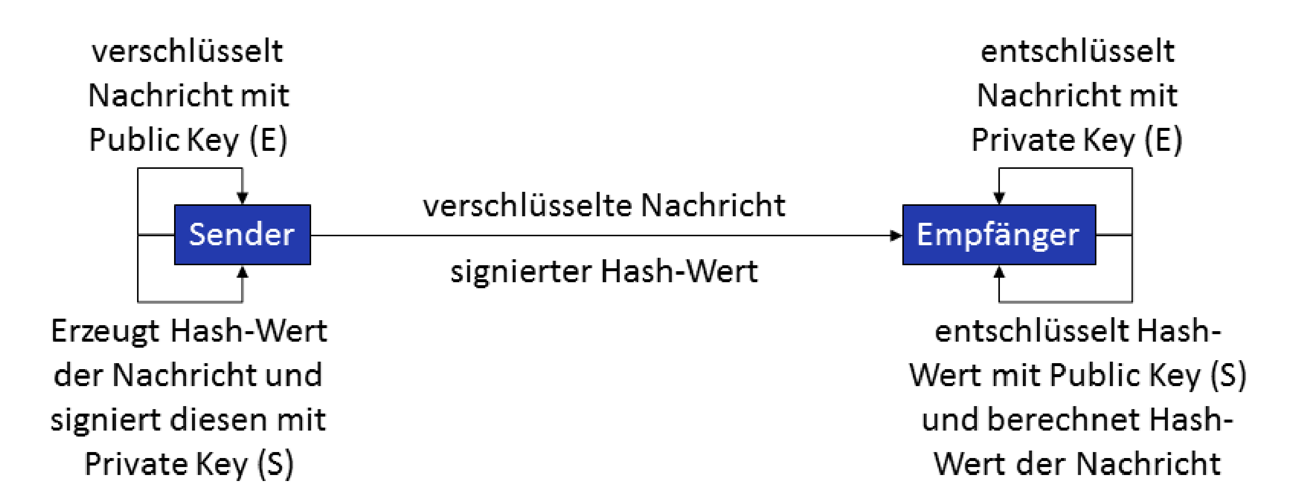


Abb. 9: asymmetrische Verschlüsselung mit digitaler Signatur

Anstatt die gesamte Nachricht zu signieren, wird der Hash-Wert der Nachricht berechnet und dieser digital signiert. Dadurch lässt sich Datenvolumen sparen. Im Anschluss werden die verschlüsselte Nachricht und der signierte Hash-Wert versendet. Der Empfänger kann nun die Nachricht mit seinem Private Key entschlüsseln und den Hash-Wert der Nachricht mit dem Public Key des Senders entschlüsseln. Wenn der Empfänger nun selbst den Hash-Wert der Nachricht berechnet und den gleichen Hash-Wert erhält, so kann er davon ausgehen, dass die Nachricht nicht kompromittiert wurde.[137]

7 Kriterienkatalog zur Produktauswahl

Autoren: Amelie Böer, Thomas Enderle, Matthias Holzwarth, Dominik Schrahe

Das folgende Kapitel ist in zwei Bereiche unterteilt. In Kapitel 7.1 wird auf den Kriterienkatalog selbst genau eingegangen. Anschließend wird der Reifegrad der ausgewählten Projekte unternehmensunabhängig bestimmt.

7.1 Erstellung des Kriterienkatalogs

Im Folgenden wird die Erstellung des Kriterienkatalogs detailliert aufgezeigt. Hierfür werden sowohl das gewählte Vorgehen als auch die gewählten Kriterien erläutert und begründet.

Gewähltes Vorgehen

Bei der Erstellung des Kriterienkatalogs wird grundlegend ein Vorgehen analog zum Navica-Golden Open Source Maturity Model[138] gewählt. Es ist laut dem Burton Group Report (Assesing OSS Projects 2005) das am besten dokumentierte und am einfachsten anwendbare

[137] Vgl. Witt, K.-U. (2014), S. 178
[138] Vgl. MALAYSIAN PUBLIC SECTOR OPEN SOURCE SOFTWARE (OSS) INITIATIVE (o. J.), S. 63 ff.

Open Source Maturity Model.[139] Stellenweise wird sich für ein hiervon abweichendes Vorgehen entschieden um speziell auf die gegebenen Anforderungen angepasstes Ergebnis zu erzielen. Dies ist an den entsprechenden Stellen vermerkt.

Die Methode des Navica-Golden Open Source Maturity Model baut auf folgenden drei Phasen auf:

1. Phase: Bewertung der Kriterien-Reifegrade
   In dieser Phase werden zunächst einzelne Kriterien festgelegt. Jedes Kriterium wird anschließend in vier Stufen bewertet:
   - Die benötigten Anforderungen werden festgelegt.
   - Benötigte Informationen werden eingeholt.
   - Reifegrad des Kriteriums wird bewertet.
   - Jedem Kriterium wird eine Punktzahl zwischen 0 und 2 zugeordnet.[140]
2. Phase: Einführung des Bewertungsfaktors
   Jedem Kriterium wird im nächsten Schritt eine Gewichtung abhängig von der Bedeutung des Aspekts zugeteilt.[141]
   Durch den eingeführten Bewertungsfaktor kann der Kriterienkatalog wichtigen Aspekten ihrer realen Bedeutung gerecht werden. Die Kriterien müssen je nach gewünschtem Einsatz der Cloud und je Unternehmen bzw. Branche unterschiedlich gewichtet werden. Die hier durchgeführte Gewichtung besitzt daher keine allgemeine Gültigkeit, sondern bezieht sich jeweils speziell auf die Situation des gezeigten Unternehmens.
3. Phase: Kalkulation des gesamten Produkt-Reifegrades
   In der letzten Phase werden die einzelnen Kriterien-Punktzahlen berechnet und das Gesamtergebnis ermittelt. Hierfür wird die Zielerreichung abhängig von der maximal möglichen Punktzahl angegeben.[142] Beim Vergleich mehrerer Produkte ist das Tool mit der höchsten Zielerreichung zu empfehlen.

[139] Vgl. Lopez Lujan, J. M. (2013), S. 33
[140] Im N-OSMM ist eine Punktzahl zwischen 0 und 10 möglich. Da viele der gewählten Kriterien nicht derart fein untergliederbar und einige nicht metrisch messbar sind, wird sich für eine gröbere Skalierung entschieden. Die gewählte Skalierung entspricht der von KOS, auf welche auch die Open Source-spezifischen Kriterien aufbauen.
[141] Im N-OSMM muss die maximal erreichbare Punktzahl 100 ergeben. Die gewählte Vorgehensweise ermöglicht hingegen eine uneingeschränkte Gewichtung mit geraden Zahlen.
[142] Die maximal mögliche Punktzahl beträgt aufgrund der offenen Gewichtung (vgl. Fußnote Nr. 141) nicht genau 100. Durch die Angabe der prozentualen Zielerreichung kann daher ein besseres Gefühl für den Erfüllungsgrad der untersuchten Produkte entwickelt werden.

Gewählte Kriterien

In Abb. 10 sind alle gewählten Kriterien aufgelistet. Im Anschluss wird die Kriterienauswahl begründet und im Kriterienkatalog genau erläutert.

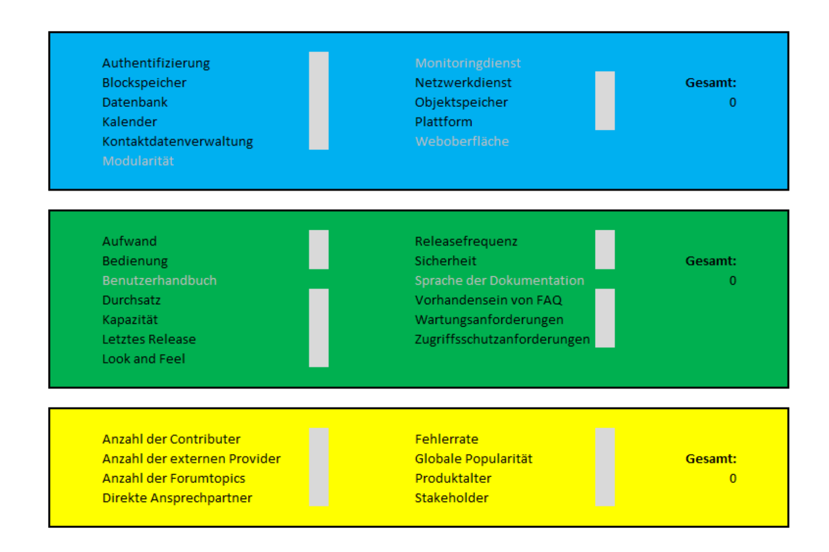


Abb. 10: Übersicht der gewählten Kriterien

Begründung der Kriterienauswahl

In der ersten Phase werden zunächst die Anforderungen ausgewählt. Der erstelle Katalog enthält 32 Kriterien, welche sich auf die funktionale, nicht-funktionale und Open Source-spezifischen Anforderungen beziehen. Bei der anschließenden Gewichtung ist dringend auf das Gewichtungsverhältnis zwischen den drei Bereichen zu achten.

<u>Auswahl der funktionalen Anforderungen:</u>

Die funktionalen Anforderungen beziehen sich auf die tatsächlichen Funktionen der Tools.

Folgende Kriterien wurden von der Quelle Golembowska, A. u. a. (2012) übernommen. Dies wird im Folgenden aufgrund der besseren Lesbarkeit mit KOS abgekürzt. Genaueres zu der Auswahl der Quelle wird bei den Open Source Kriterien erläutert:

- Plattform
- Modularität

Zusätzlich wurden die folgenden Kriterien ausgewählt, welche sich aus den angebotenen Funktionalitäten der untersuchten Produkte zusammensetzen:

- Authentifizierung
- Blockspeicher
- Datenbank
- Kalender

- Kontaktdatenverwaltung
- Monitoringdienst
- Netzwerkdienst
- Objektspeicher
- Weboberfläche

Auswahl der nicht-funktionalen Anforderungen:

Für die Auswahl von Softwaresystemen ist es neben den funktionalen Anforderungen, die wie oben beschrieben festlegen, was ein Produkt tun soll, die Beleuchtung von nicht-funktionalen Anforderung essentiell. Diese legen die Eigenschaften des Produktes fest.

Für die Ermittlung nicht-funktionaler Anforderung gibt es Standards sowie vorgegebene Normen und Muster. Behandelt werden in diesem Kapitel die deutsche Norm DIN 66272, der US-amerikanische Standard IEEE 830-1998 sowie die Materialiensammlung Volere.

Die angesprochene DIN 66272 kann für die geforderte Aufgabenstellung angewandt werden, hierbei ist allerdings zu beachten, dass die letzte Version der Norm aus dem Jahr 1994 ist und dies mittlerweile ersatzlos gestrichen wurde. Es ist daher empfehlenswert, einen alternativen Leitfaden zu betrachten.

Eine weitere, aktuell gültige, Gliederung sind die Software Requirement Specifications des IEEE Standards 830-1998.[143] Diese liefert Vorschläge für die Einbeziehung nicht-funktionaler Anforderungen in ein Softwaresystem und kann für die Beurteilung der Verwendung von Clouds Anwendung finden. Es ist dabei allerdings zu beachten, dass die aktuell noch gültige Fassung aus dem Jahr 1998 stammt.[144] Da dies im Bereich der Softwaretechnik eine vergleichbar lange Zeit ist, darf davon ausgegangen werden, dass die dort aufgeführten Punkte nicht mehr heutigen Anforderungen entsprechen. Es ist aus diesem Grund zu empfehlen, dass das Volere Requirements Specification Template (auch als Volere Template bezeichnet) zum Einsatz kommt, welches im nachfolgenden Abschnitt beschrieben wird.

Das Volere Template ist ein durchdachter Leitfaden für die Erstellung von Anforderungen. Es werden dabei auch die unterschiedlichen Interessen einzelner Stakeholder berücksichtigt.[145] Zwar gibt es eine genaue Beschreibung, wie ein Volere Template zu erstellen und benutzen ist[146], auf eine Erklärung oder Anwendung wird in dieser Ausarbeitung allerdings verzichtet. Da es das Hauptanliegen dieses Kapitels ist, nicht-funktionale Anforderungen für die Auswahl von Clouds zu erhalten, genügt das Extrahieren dieser Punkte.

[143] Vgl. Zehnter, C./Burger, A./Ovtcharova, J. (2012), S. 49
[144] Vgl. Balzert, H. (2011), S. 486
[145] Vgl. Zehnter, C./Burger, A./Ovtcharova, J. (2012), S. 50
[146] Vgl. ebenda, S. 50

Die im Volere Template genannten nicht-funktionalen Anforderungen sind:

- Benutzbarkeitsanforderung
- Durchsatz
- Kapazität
- Kulturelle Anforderungen
- Look and Feel
- Operationale Anforderung
- Performance
- Rechtliche Anforderung
- Sicherheit
- Wartungsanforderungen
- Zugriffsschutzanforderungen

Für folgende drei Bereiche wird eine Änderung vorgenommen:

1. Aufgrund der hohen Bedeutung des Kriteriums Sicherheit wird dieses erweitert. Wichtig ist bei der Auswahl der Kriterien, dass diese konkret und, falls möglich, messbar sind. Es werden folgende Kriterien benutzt:

   - Releasefrequenz
   - Letztes Release

   Das Kriterium Sicherheit wird im Katalog trotzdem beibehalten. Es sollen, wie im späteren Verlauf dieser Arbeit beschrieben wird, die allgemeinen Anforderungen von Unternehmen an die Sicherheit der von ihnen verwendeten Software – im Speziellen die Cloud – repräsentiert werden.

2. Der Bereich der Benutzbarkeitsanforderungen ist aufgrund seiner direkten Auswirkung auf die Benutzer von Bedeutung. Das Kriterium wird wie folgt aufgespalten:

   - Benutzerhandbuch
   - Sprache der Dokumentation
   - Vorhandensein von Frequently Asked Questions (FAQ)

3. Im Kapitel 3 wird auf die besonderen Anforderungen an Versicherungsunternehmen im deutschen Kulturkreis eingegangen. Diese beziehen sich in erster Linie auf einen hohen Sicherheitsstandard. Aus diesem Grund wird für den Bereich der kulturellen

Anforderungen kein weiteres Kriterium definiert.

4. Aktuell sind neben den Sicherheitsbestimmungen keine weiteren rechtlichen Anforderungen bekannt. Aus diesem Grund wird für diesen Bereich kein Kriterium definiert.

5. Der Punkt operationale Anforderungen der Volere-Sammlung wurde in seine Unterpunkte, die Wartungsanforderung und die Zugriffsschutzanforderung, aufgeteilt. Diese werden im Kriterienkatalog separat beurteilt.

Alle weiteren Kriterien werden beibehalten.

Auswahl der Open Source-spezifischen Anforderungen:

Der Open Source-spezifische Bereich wird in drei Stufen entwickelt:

1. Auswahl eines Kriterienkatalogs:

   Zunächst muss eine Liste möglicher Kriterien ausgewählt werden. Da im Bereich von Open Source-Produkten bereits eine Vielzahl von branchenunabhängigen Kriterienkatalogen existiert, wird ein solcher gewählt und anschließend angepasst. Als Grundlage wird KOS[147]aufgrund der guten Messbarkeit der Kriterien gewählt.

2. Filterung nach relevanten Kriterien

   Im zweiten Schritt wird eine Einschränkung durch das Projektteam vorgenommen. Hierbei wird auf folgende zwei Aspekte Wert gelegt:

   - Kriterien müssen auf den Cloud-Bereich anwendbar sein.
   - Kriterien dürfen sich nicht mit den Bereichen funktional und nicht-funktional überschneiden.

   Es entsteht die Notwendigkeit, folgende Kriterien zu entfernen:

   - Lizenzen (Bereich Produkt):

     Dieses Kriterium wird entfernt, da es bereits ein eigenes Kapitel hierzu gibt. Die Eignung einer Lizenz hängt nicht allein von der Lizenz selbst, sondern in erster Linie von der gewünschten Anwendung ab.

   - Programmiersprache (Bereich Produkt):

     Die Programmiersprache ist nur für PaaS relevant. Es werden nur IaaS-Produkte verglichen und das Kriterium wird daher entfernt.

   - Funktionalität (Bereich Produkt):

[147] Vgl. Golembowska, A. u. a. (2012), S. 204 ff.

Dieses Kriterium entspricht dem Bereich funktionale Kriterien und kann somit entfernt werden.

- Kompatibilität (Bereich Produkt):

  Bei dem Kriterium Kompatibilität soll der Benutzer alle bereits vorhandenen Anwendungen angeben, mit denen das neue Tool kompatibel sein muss. Da die Kriterien des entwickelten Kriterienkatalogs allerdings nicht flexibel sein sollen, wird dieses Kriterium entfernt. Der Benutzer sollte dies bereits vor Anwendung des Kriterienkatalogs überprüfen.

- Codedokumentation (Bereich Dokumentation):

  Der bereits vorhandene Code muss dokumentiert sein. Hierbei geht es nicht allein um das reine vorhanden sein einer Codedokumentation, sondern auch um deren Qualität. Die Dokumentation muss auch ohne Kenntnis der Codes verständlich sein. Dieses Kriterium ist besonders bedeutend für eine geplante Weiterentwicklung, da ansonsten von einem sehr hohen Mehraufwand auszugehen ist. Ist keine oder eine nicht ausreichend gute Dokumentation vorhanden, können zudem bei der Weiterentwicklung entstehende Fehler schlechter lokalisiert werden. Es ist allerdings ohne eine vollständige Sichtung des Codes nicht feststellbar, ob diese durchgehend vorhanden ist und ob deren Qualität ausreicht. Da eine vollständige Codesichtung im Rahmen dieser Arbeit nicht möglich ist, wird dieses Kriterium entfernt.

- Qualität der Dokumentation (Bereich Dokumentation):

  Bei diesem Aspekt wird die Qualität aller Dokumentationen bewertet. Dies beinhaltet sowohl die Codedokumentation, als auch das Benutzerhandbuch. Die Qualität der jeweiligen Dokumentation wird in dieser Ausarbeitung allerdings direkt bei den angesprochenen Kriterien mitbewertet. Daher wird der Aspekt Qualität der Dokumentation entfernt.

- Anzahl der Downloads (Bereich Marktakzeptanz):

  Private Clouds werden für gewöhnlich nicht über den Downloadanbieter chip.de heruntergeladen. Daher kann das Kriterium nicht wie vorgeschlagen gemessen werden und wird entfernt.

- Deutschlandweite Popularität (Bereich Marktakzeptanz):

  Im Gegensatz zur globalen Popularität spiegelt die deutschlandweite Popularität die Akzeptanz auf dem deutschen Markt wieder. Dieser Wert kann

laut KOS[148] abermals über die Seite Alex.com abgefragt werden. Die deutschlandweite Popularität besitzt gerade für deutsche Versicherungsunternehmen eine hohe Relevanz. Auslöser hierfür sind zum einen der Aspekt des hohen Datenschutzes in Deutschland und dem Bekanntheitsgrad bei den deutschen Kunden. Das Kriterium muss dennoch entfernt werden, da Alex.com nur bei Seiten der Domaine .de den deutschen Traffic Rank anzeigt – die Ermittlung der deutschlandweiten Popularität ist für die untersuchten Tools daher nicht möglich.

- Platzierung in com! Top 100 der Open Source Projekte (Bereich Marktakzeptanz):

  Dieser Aspekt wird entfernt, da dieses Kriterium für den Private Cloud-Bereich nicht aussagekräftig ist. Keines der gewählten Produkte befindet sich auf der Liste. Zudem wurde diese Liste 2012 erstellt und besitzt somit keine aktuelle Aussagekraft.

- Marktanteile auf Basis von Fachpublikationen (Bereich Marktakzeptanz):

  Dieses Kriterium eindeutig messbarer Wert für die Verbreitung des jeweiligen Open Source-Produkts. Alle zu untersuchenden Open Source-Produkte gehören in diesem Fall allerdings dem gleichen Bereich an. Es gibt aktuell keine Fachpublikation über die entsprechenden Marktanteile. Aus diesem Grund wird das Kriterium Marktanteile auf Basis von Fachpublikationen entfernt.

Folgende Kriterien werden dem nicht-funktionalen Bereich zugeordnet, da diese auch bei kommerziellen Produkten wichtig sind:

- Benutzerhandbuch
- Letztes Release[149]
- Sprache der Dokumentation
- Vorhandensein von FAQ

Bei folgenden Aspekten handelt es sich um funktionale Kriterien:

- Plattform
- Modularität

[148] Vgl. Golembowska, A. u. a. (2012), S. 210
[149] Dieses Kriterium wird in KOS als Releaseabstände bezeichnet. Aufgrund der irreführenden Bezeichnung wird es umbenannt.

Alle weiteren Kriterien werden übernommen.

3. Erweiterung um zusätzliche Kriterien:

   Im Bereich der Marktakzeptanz musste eine Vielzahl von Kriterien entfernt werden. Aus diesem Grund wird das Kriterium Anzahl der Contributer hinzugefügt.

Ausführlicher Kriterienkatalog

Anbei befindet sich eine alphabetische Übersicht der ausgewählten Kriterien. Diese beinhaltet jeweils eine Beschreibung des Kriteriums, die Motivation für das Kriterium und eine Hilfestellung zur Bewertungsdurchführung. Dieser Bereich sollte bei der Durchführung der unternehmensspezifischen Gewichtung vorliegend sein und als Nachschlagwerk dienen.

Anzahl der Contributer

Bereich: Open Source

Beschreibung: Über die Code-Reposity-Seite github.com wird die Anzahl der Contributer gemessen.

Motivation: Eine hohe Contributeranzahl ist ein Indiz für eine hohe Qualität und bietet die Möglichkeit einer schnellen Fehlerbereinigung.

Hinweise zur Bewertungsdurchführung: Eine hohe Contributer-Anzahl wird bevorzugt.

- 0 = unter 100 Contributers
- 1 = zwischen 100 und 200 Contributers
- 2 = über 200 Contributers

Anzahl der externen Provider

Bereich: Open Source

Beschreibung: Dieses Kennzeichen misst, ob und wie viele externe Provider individuellen Support anbieten.

Motivation: Durch externe Provider kann ein Unternehmen den Einsatz / die Wartung an einen Dienstleister abgeben und benötigt keine eigenen Experten.

Hinweise zur Bewertungsdurchführung: Die von KOS[150] vorgeschlagene Skalierung wird übernommen. Bei mehr als einem externen Provider ist das Unternehmen nicht den Preisvorstellungen des Anbieters unterworfen. Daher wird hierfür die höchste Punktzahl gewählt.

Hinweise zur Bewertungsdurchführung:

[150] Vgl. Golembowska, A. u. a. (2012), S. 209

- 0 = keine externen Provider
- 1 = Verfügbarkeit eines externen Providers
- 2 = Mindestens zwei externe Provider verfügbar

Anzahl der Forumtopics

Bereich: Open Source

Beschreibung: Messung der Topics des jeweiligen Cloudforums. Nur Foren, auf welche von der offiziellen Seite aus verwiesen wird, werden beachtet.

Motivation: Der Aspekt Anzahl der Forumtopics ist ein Indiz für bereits diskutierte Themen in der offiziellen Community. Bei einer hohen Anzahl ist die Wahrscheinlichkeit hoch, mit geringem Aufwand schnell eine Antwort auf eine Frage zu finden.

Hinweise zur Bewertungsdurchführung: Die von KOS[151] vorgeschlagene Skalierung wird übernommen.

- 0 = unter 10.000 Forentopics
- 1 = 10.000 – 100.000 Forentopics
- 2 = über 100.000 Forentopics

Aufwand

Bereich: Nicht-funktional

Beschreibung: Es wird der Aufwand beschrieben, der notwendig ist um mit einer Cloud-Lösung zu arbeiten und sie aufrecht zu halten. Verwendet wird für die Beurteilung die im späteren Kapitel 8 beschriebene Implementierung sowie Testberichte.

Motivation: Eine Lösung kann nur dann effektiv eingesetzt werden, wenn sich der Aufwand gering hält. Nur dann ist der Einsatz gewährleistet.

Hinweise zur Bewertungsdurchführung:

- 0 = Schlechteste Bewertung in Testberichten
- 1 = Dazwischenliegende Lösungen
- 2 = Beste Bewertung in Testberichten

Authentifizierung

Bereich: Funktional

Beschreibung: Es wird überprüft, ob das Tool eine Authentifizierung ermöglicht.

[151] Vgl. Golembowska, A. u. a. (2012), S. 208

Motivation: Je nach Authentifizierungsmöglichkeit wird eine höhere Sicherheit erreicht.

Hinweise zur Bewertungsdurchführung:

- 0 = Authentifizierung wird nicht ermöglicht.
- 1 = Ein-Faktor-Authentifizierung wird ermöglicht.
- 2 = 2FA wird ermöglicht.

Bedienung

Bereich: Nicht-funktional

Beschreibung: Es wird die Bedienungsfreundlichkeit beurteilt. Verwendet wird für die Beurteilung zum einen die im späteren Kapitel 0 beschriebene Implementierung. Zusätzlich werden Testberichte zu Rate gezogen.

Motivation: Eine Lösung kann nur dann effektiv eingesetzt werden, wenn eine benutzerfreundliche Bedienung vorhanden ist. Dies soll im Rahmen dieser Arbeit ausgewertet werden.

Hinweise zur Bewertungsdurchführung:

- 0 = Schlechteste Bewertung in Testberichten
- 1 = Dazwischenliegende Lösungen
- 2 = Beste Bewertung in Testberichten

Benutzerhandbuch

Bereich: Nicht-funktional

Beschreibung: Es wird ein Benutzerhandbuch benötigt. Hierbei ist zunächst eine gute Verständlichkeit von Bedeutung. Zusätzlich zur Bedienung umfasst das Handbuch auch eine Installationsanleitung.

Motivation: Ist kein ausreichendes Benutzerhandbuch vorhanden, entsteht Mehraufwand bei der Installation und Bedienung. Zudem können etwaige Features des Produkts unentdeckt und somit ungenutzt bleiben.

Hinweise zur Bewertungsdurchführung:

- 0 = kein Benutzerhandbuch vorhanden
- 1 = Benutzerhandbuch nur für Installation oder Bedienung vorhanden. / Benutzerhandbuch besitzt keine gute Qualität.
- 2 = Benutzerhandbuch umfasst Installation und Bedienung und hat zudem eine gute Qualität.

Blockspeicher

Bereich: Funktional

Beschreibung: Es wird überprüft, ob das Tool einen Blockspeicher zur Verfügung stellt.

Motivation: Blockspeicher werden mit einem herkömmlichen Dateisystem formatiert und dienen in IaaS-Umgebungen in der Regel zur Bereitstellung von Betriebssystemen inklusive Anwendungen. Zum Ausführen dieser, werden vorhandene Blockspeicher-Volumes in einer Virtuellen Maschine eingebunden. Für die Dateispeicherung sind Blockspeicher weniger geeignet.

Hinweise zur Bewertungsdurchführung:

- 0 = Blockspeicher wird nicht zur Verfügung gestellt.
- 2 = Blockspeicher wird zur Verfügung gestellt.

Datenbank

Bereich: Funktional

Beschreibung: Es wird überprüft, ob das Tool eine Datenbank zur Verfügung stellt.

Motivation: Benötigt das Unternehmen für die gewünschte Benutzung des Tools eine Datenbank, kann dies durch entsprechende Gewichtung abgefragt werden.

Hinweise zur Bewertungsdurchführung:

- 0 = Datenbank wird nicht zur Verfügung gestellt.
- 2 = Datenbank wird zur Verfügung gestellt.

Direkte Ansprechpartner

Bereich: Open Source

Beschreibung: Das Kennzeichen gibt an, ob auf der offiziellen Open-Source Homepage eine Kontaktmöglichkeit zu einem Experten vorhanden ist.

Motivation: Ist das Kriterium erfüllt, ist eine einfache Möglichkeit geboten Hilfe für individuelle Probleme zu erhalten.

Hinweise zur Bewertungsdurchführung: Die von KOS[152] vorgeschlagene Skalierung wird übernommen. Es kann nur zwischen vorhanden und nicht vorhanden entschieden werden.

- 0 = vorhanden
- 2 = nicht vorhanden

[152] Vgl. Golembowska, A. u. a. (2012), S. 208

Durchsatz

Bereich: Nicht-funktional

Beschreibung: Die Thematik des Durchsatzes wird vom später beschriebenen Kriterium Performance abgeleitet. Ziel ist es, einen möglichst hohen Durchsatz zu erreichen, da ein geringer Wert für Unternehmen nicht praktikabel ist.[153] Genutzt werden für die Beurteilung dieses nicht-funktionalen Kriteriums Testberichte.

Motivation: Durchsatz ist von Bedeutung, da Unternehmen schnelle Reaktionszeiten benötigen, um einen effektiven Einsatz einer Lösung zu gewährleisten.[154]

Hinweise zur Bewertungsdurchführung:

- 0 = Schlechteste Bewertung in Testberichten
- 1 = Dazwischenliegende Lösungen
- 2 = Beste Bewertung in Testberichten

Fehlerrate[155]

Bereich: Open Source

Beschreibung: Bei der Fehlerrate wird die durchschnittliche monatliche Fehlerrate für das Jahr 2014 gemessen.[156] Dieses Kennzeichen wird über das jeweilige Bug Tracking Tool bestimmt.

Motivation: Die Fehlerrate spiegelt die Fehleranfälligkeit eines Projekts wieder.

Hinweise zur Bewertungsdurchführung: Die von KOS[157] vorgeschlagene Skalierung wird übernommen.

- 0 = durchschnittlich mehr als 1.000 Fehler im Monat
- 1 = durchschnittliche zwischen 50 und 1.000 Fehlern
- 2 = durchschnittlich unter 50 Fehlern

Globale Popularität

Bereich: Open Source

[153] Vgl. Cardy, R. L./Leonard, B. (2011), S. 4 ff.
[154] Vgl. ebenda, S. 4 ff.
[155] Dieses Kriterium wird sehr kritisch betrachtet. Eine äußerst geringe Fehlerrate kann auch ein Indiz für Inaktivität der Community oder eine schlechte Fehlersuche sein. Das Kriterium wird beibehalten. Die angesprochene Kritik wird allerdings bei der Empfehlung für die Gewichtung berücksichtigt.
[156] Die ursprünglich vorgeschlagene Messung dieses Kriteriums wird nicht übernommen. Statt eine Betrachtung von nur einem Monat wird ein Durchschnittswert von zwölf Monaten genommen, um die allgemeine Gültigkeit des Kriteriums zu erhöhen. Hierdurch werden Schwankungen, beispielsweise aufgrund von Ferienzeiten, ausgeglichen.
[157] Vgl. Golembowska, A. u. a. (2012), S. 206

Beschreibung: Die Popularität der jeweiligen Cloud kann über die durchschnittliche Anzahl von Pageviews über einen Zeitraum von drei Monaten ermittelt werden. Hierbei wird wie in KOS vorgeschlagen die Seite Alexa.com benutzt.[158]

Motivation: Die globale Popularität ist ein Kennzeichen für die Marktakzeptanz und wirkt sich auf die Beständigkeit eines Open Source Produkts aus.

Hinweise zur Bewertungsdurchführung: Bei diesem Kriterium wird nicht die vorgeschlagene Skalierung übernommen, da sich die ausgewählten Cloud-Produkte in einem anderen Bereich bewegen. Stattdessen werden die Produkte miteinander verglichen.

- 0 = schlechtester Traffic Rank
- 1 = andere Traffic Ranks
- 2 = bester Traffic Rank

Kalender

Bereich: Funktional

Beschreibung: Es wird überprüft, ob das Tool einen Kalender zur Verfügung stellt.

Motivation: Benötigt das Unternehmen für die gewünschte Benutzung des Tools einen Kalender, kann dies durch entsprechende Gewichtung abgefragt werden.

Hinweise zur Bewertungsdurchführung:

- 0 = Kalender wird nicht zur Verfügung gestellt.
- 2 = Kalender wird zur Verfügung gestellt.

Kapazität

Bereich: Nicht-funktional

Beschreibung: Die Kapazität ist abhängig von dem Server, auf dem die Cloud-Lösung gespeichert wird. Theoretische Überlegungen können aber in Bezug auf die Skalierbarkeit gemacht werden. Da hierbei die Auswirkung auf das Leistungsverhalten durch das Hinzufügen von Ressourcen beurteilt wird.[159]

Motivation: Ähnlich wie bei der Performance ist eine große Skalierbarkeit unersetzlich für den Einsatz im Businessumfeld, deshalb ist eine Aufnahme in den Kriterienkatalog notwendig.

Hinweise zur Bewertungsdurchführung:

- 0 = Geringstmögliche Skalierbarkeit

[158] Vgl. Golembowska, A. u. a. (2012), S. 210
[159] Vgl. Fowler, M. (2003), S. 23

- 1 = Dazwischenliegende Lösungen
- 2 = Bestmöglichste Skalierbarkeit

Kontaktdatenverwaltung

Bereich: Funktional

Beschreibung: Es wird überprüft, ob das Tool eine Kontaktdatenverwaltung zur Verfügung stellt.

Motivation: Benötigt das Unternehmen für die gewünschte Benutzung des Tools eine Kontaktdatenverwaltung, kann dies durch entsprechende Gewichtung abgefragt werden.

Hinweise zur Bewertungsdurchführung:

- 0: Kontaktdatenverwaltung wird nicht zur Verfügung gestellt.
- 2: Kontaktdatenverwaltung wird zur Verfügung gestellt.

Letztes Release

Bereich: Nicht-funktional

Beschreibung: Das Kriterium Letztes Release misst die verstrichene Zeit seit der letzten Version. Hierunter fallen jegliche Arten von Updates, so auch Hotfixes. Dies entspricht dem Kriterium Releaseabstände von KOS.[160]

Motivation: Durch Releases werden nicht nur neue Funktionen etabliert, sondern auch Sicherheitslücken ausgebessert. Bei sehr langen Releaseabständen ist nicht gewährleistet, dass das Produkt sicherheitstechnisch auf einem aktuellen Stand ist.

Hinweise zur Bewertungsdurchführung: Die von KOS[161] vorgeschlagene Skalierung wird übernommen.

- 0 = letztes Release war vor über fünf Monaten
- 1 = letztes Release ist zwischen einem und fünf Monaten her
- 2 = letztes Release war vor unter einem Monat

Look and Feel

Bereich: Nicht-funktional

Beschreibung: Dieses Kennzeichen gibt den ersten optischen Eindruck wieder. Es wird der subjektive Ersteindruck des Erscheinungsbilds bewertet. Hierzu werden die Grundlagen der

[160] Vgl. Golembowska, A. u. a. (2012), S. 206 f.
[161] Vgl. ebenda, S. 206 f.

Softwareergonomie, welche in der DIN-Norm 9241 festgehalten sind.[162] Hierbei ist vor allem die Thematik der Lernförderlichkeit von Bedeutung.[163]

Motivation: Für die Akzeptanz des neuen Systems ist es wichtig, dass die Handhabung und schnelle Lernfähigkeit besteht. Dies ist besonders, wie bereits in Kapitel 3 dargestellt, im Versicherungsgewerbe notwendig, da hier in den IT-Abteilungen eine tendenziell traditionelle Vorstellung herrscht, muss Look and Feel in die Bewertung mit einfallen.

Hinweise zur Bewertungsdurchführung: Die Bewertung von Look and Feel wird anhand von öffentlich zugänglichem Material der Cloud-Lösungen durchgeführt.

- 0 = Schlechtester Look-and-Feel-Eindruck
- 1 = Dazwischenliegende Lösungen
- 2 = Bester Look-and-Feel-Eindruck

Modularität

Bereich: Funktional

Beschreibung: Es wird überprüft, ob das Tool modular aufgebaut ist.

Motivation: Modularität ermöglicht eine Erweiterung des Produktes um benötigte Bereiche. Zudem wirkt sich hierbei eine Änderung in einem Modul nicht auf alle Bereiche aus.

Hinweise zur Bewertungsdurchführung: Es wird die von KOS[164] vorgeschlagene Skalierung übernommen.

- 0: Modularität nicht vorhanden.
- 2: Modularität vorhanden.

Monitoringdienst

Bereich: Funktional

Beschreibung: Es wird überprüft, ob das Produkt einen Monitoringdienst zur Verfügung stellt

Motivation: Wird ein Monitoringdienst benötigt, kann dies über eine entsprechende Gewichtung abgefragt werden.

Hinweise zur Bewertungsdurchführung:

- 0 = Monitoringdienst wird nicht zur Verfügung gestellt.
- 2 = Monitoringdienst wird zur Verfügung gestellt.

[162] Vgl. Kaschuba, A. (2013), S. 2 f.
[163] Vgl. ebenda, S. 2 f.
[164] Vgl. Golembowska, A. u. a. (2012), S. 206

Netzwerkdienst

Bereich: Funktional

Beschreibung: Es wird überprüft, ob das Tool einen Netzwerkdienst zur Verfügung stellt.

Motivation: Benötigt das Unternehmen für die gewünschte Benutzung des Tools einen Netzwerkdienst, kann dies durch entsprechende Gewichtung abgefragt werden.

Hinweise zur Bewertungsdurchführung:

- 0 = Netzwerkdienst wird nicht zur Verfügung gestellt.
- 2 = Blockspeicher wird zur Verfügung gestellt.

Objektspeicher

Bereich: Funktional

Beschreibung: Es wird überprüft, ob das Tool einen Objektspeicher zur Verfügung stellt.

Motivation: Objektspeicher dienen zur Ablage von Dateien, wobei diese nicht in einem klassischen (hierarchischen) Dateisystem erfolgt, das einen Indexeintrag für jede gespeicherte Datei benötigt. Bei Objektspeichern werden Dateien ähnlich einem Key-Value Store abgelegt, wobei der Schlüssel den eindeutigen Zugriffsnamen auf die Datei (Value) darstellt. Diese Art der Datenspeicherung macht sich besonders bei der Ablage sehr großer Dateimengen bemerkbar, da Objektspeicher sehr viel besser skalieren als herkömmliche Speicherverfahren.

Hinweise zur Bewertungsdurchführung:

- 0 = Objektspeicher wird nicht zur Verfügung gestellt.
- 2 = Objektspeicher wird zur Verfügung gestellt.

Plattform

Bereich: Funktional

Beschreibung: Es wird überprüft, auf welchen Plattformen und Hypervisoren das Produkt lauffähig ist.

Motivation: Das Tool sollte die Plattformen/Hypervisoren unterstützen, welche in den meisten Firmen gängig sind.

Hinweise zur Bewertungsdurchführung:

- 0 = Unterstützt nicht die gängigen Plattformen und Hypervisoren
- 1 = Unterstützt nicht die gängigen Plattformen und Hypervisoren
- 2 = Hervorstechende Besonderheiten

Produktalter

Bereich: Open Source

Beschreibung: Das Kriterium Produktalter beschreibt, wie lange sich die jeweilige Cloud bereits auf dem Markt befindet.

Motivation: Befindet sich ein Produkt bereits länger auf dem Markt, kann davon ausgegangen werden, dass es sich nicht um einen kurzfristigen Trend handelt und das Produkt nach abklingen diesen wieder vom Markt verschwindet.

Hinweise zur Bewertungsdurchführung: Die von KOS[165] vorgeschlagene Skalierung wird nicht übernommen, da Cloud Projekte im Durchschnitt jünger als die dort untersuchten Open Source Projekte sind. Es wird der jeweilige Beginn des Open Source Projekts benutzt, nicht die erste stabile Version.

- 0 = weniger als zwei Jahre
- 1 =zwischen zwei und fünf Jahren
- 2 = mehr als fünf Jahre

Releasefrequenz

Bereich: Nicht-funktional

Beschreibung: Unter dem Kriterium Releasefrequenz wird die verstrichene Zeit zwischen den einzelnen Releases verstanden.

Motivation: Ähnlich wie das Kriterium Letztes Release wird aus Sicherheitsaspekten ein niedriger Abstand bevorzugt.

Hinweis zur Bewertungsdurchführung: Falls es keine festen Releaseabstände gibt, wird der Durchschnittswert der letzten zwei Jahre genutzt.

- 0 = höchste Releasefrequenz
- 1 = mittlere Lösungen
- 2= niedrigste Releasefrequenz

Sicherheit

Bereich: Nicht-funktional

Beschreibung: Auf die Sicherheit wurde bereits mehrfach eingegangen. Das Kriterium der Sicherheit vergleicht allgemein die Sicherheit der verschiedenen Cloud-Lösungen, welche in Testberichten festgehaltenen wurde.

[165] Vgl. Golembowska, A. u. a. (2012), S. 204

Motivation: Wie im Teil 2.1 wird besonders bei Versicherungen Wert auf die Sicherheit gelegt. Aus diesem Grund wird dieses nicht-funktionale Kriterium noch einmal separat aufgenommen.

Hinweise zur Bewertungsdurchführung:

- 0 = Schlechteste Bewertung in Testberichten
- 1 = Dazwischenliegende Lösungen
- 2 = Beste Bewertung in Testberichten

Sprache der Dokumentation

Bereich: Nicht-funktional

Beschreibung: Die Dokumentation muss in Sprache verfasst sein, welche die Mitarbeiter im Unternehmen sprechen. Es wird von deutschen Versicherungsunternehmen ausgegangen, bei welchem die Mitarbeiter Englischkenntnisse besitzen. Eine Dokumentation in Englisch ist zudem sinnvoll, sollte die Cloud von Mitarbeitern / Dienstleistern aus dem Ausland betreut werden.

Motivation: Eine bereits in der „richtigen“ Sprache verfasste Dokumentation erhöht das Verständnis bei den Benutzern erheblich. Ist die Dokumentation in einer Sprache verfasst, welche die Mitarbeiter nicht fließend beherrschen, muss die Dokumentation zunächst übersetzt werden bzw. könnte es weitreichende Probleme in der Anwendung geben.

Hinweise zur Bewertungsdurchführung: Ist bei einem Produkt keine Dokumentation vorhanden, wird dies bereits bei dem Kriterium Benutzerhandbuch berücksichtigt. Um eine Doppelbewertung zu vermeiden, wird in diesem Fall die Gewichtung des Kriteriums Sprache der Dokumentation manuell für das betroffene Produkt auf 0 gesetzt.

- 0 = Dokumentation weder in Deutsch noch in Englisch vorhanden.
- 1 = Dokumentation in Deutsch oder Englisch vorhanden.
- 2 = Dokumentation ist sowohl in Englisch als auch in Deutsch vorhanden.

Stakeholder

Bereich: Open Source

Beschreibung: Das Kriterium Stakeholder misst die Einzelpersonen/Unternehmen, welche das jeweilige Projekt aufgrund von langfristigen Interessen unterstützen (vgl. Quelle KOS)

Motivation: Unterstützende Einzelpersonen/Unternehmen tragen zur Beständigkeit des Produkts auf dem Markt bei.

Hinweise zur Bewertungsdurchführung: Die von KOS[166] vorgeschlagene Skalierung wird übernommen.

- 0 = keine Stakeholder
- 1 = zwischen einem und zehn Stakeholdern
- 2 = über zehn Stakeholder

Wartungsanforderungen

Bereich: Nicht-funktional

Beschreibung: Beim Einsatz von Software ist stets zu beachten, dass die Wartung einen festen und wiederkehrenden Bestandteil darstellt. Hierfür sind zum einen zeitliche Ressourcen aufzuwenden, um die Wartung zu ermöglichen. Weiterhin ist, im Interesse des Fortbestandes der Open-Source-Cloud-Lösung, zu beachten, dass eine Wartung einfach durchzuführen ist.

Motivation: Es soll verhindert werden, dass eine Lösung wegen zu problematischer Wartungsanforderungen nach Inbetriebnahme nicht weiter eingesetzt wird.

Hinweise zur Bewertungsdurchführung:

- 0 = Umständlichste Wartbarkeit
- 1 = Dazwischenliegende Lösungen
- 2 = Einfachste Wartbarkeit

Weboberfläche

Bereich: Funktional

Beschreibung: Es wird bewertet, ob das Produkt eine Weboberfläche zur Verfügung stellt.

Motivation: Eine Weboberfläche stellt eine einfache Zugriffsmöglichkeit auf die Funktionen einer Anwendung zur Verfügung. Hierüber kann die entsprechende Anwendung ohne Installation eines Client-Programms genutzt werden. Dies ermöglicht die Benutzung der Anwendung auf externen Rechnern und mobilen Geräten.

Hinweise zur Bewertungsdurchführung:

- 0 = Weboberfläche wird nicht zur Verfügung gestellt.
- 2 = Weboberfläche wird zur Verfügung gestellt.

[166] Vgl. Golembowska, A. u. a. (2012), S. 207

Zugriffsschutzanforderungen

Bereich: Nicht-funktional

Beschreibung: Zugriffsschutz ist eine notwendige Maßnahme, um unerwünschten Zugriff auf die in der Cloud hinterlegten Daten zu verhindern.

Motivation: Bereits im Kapitel 3 wurde darauf hingewiesen, dass Versicherungen mit Reputationsverlusten konfrontiert sind, sobald mit Störungen – zu denen auch der Zugriff auf Daten durch Unberechtigte gehört – zu rechnen ist. Da dies auch auf andere Branchen zutrifft, wird dieser Punkt beschrieben.

Hinweise zur Bewertungsdurchführung:

- 0 = Niedrigste Anforderungen an den Zugriffschutz
- 1 = Dazwischenliegende Lösungen
- 2 = Höchste Anforderungen an den Zugriffsschutz

7.2 Bewertung der Kriterien-Reifegrade

Wie in Kapitel 7.1 erläutert, muss für jedes Produkt der Reifegrad für die einzelnen Kriterien bestimmt werden.

Bewertung des Tools Eucalyptus

Funktionale Kriterien:

Eucalyptus benötigt Hypervisoren. Hierbei kommen Xen und KVM (Kernel-based Virtual Maschine) in Frage.[167] Es wird bei Plattform hierfür ein Punkt vergeben.

Eucalyptus selbst bietet keinen Monitoringdienst an. Da allerdings auf der offiziellen Webseite das Open Source Monitoringtool Nagios inklusive der Installationsanleitung beschrieben wird, wird das Kriterium dennoch mit erfüllt und daher mit zwei Punkten bewertet.[168]

Auch eine Weboberfläche ist bei Eucalyptus vorhanden.[169] Dies entspricht zwei Punkten.

Eucalyptus ist modular aufgebaut[170] und erhält hierfür zwei Punkte.

Objektspeicher[171], Blockspeicher[172], Datenbank[173] und Netzwerkdienst[174] werden von Eucalyptus zur Verfügung gestellt und jeweils mit zwei Punkten bewertet.

[167] Vgl. Baun, C. u. a. (2011), S. 94
[168] Vgl. op5 (o. J.)
[169] Vgl. Büst, R. (2012)
[170] Vgl. Eucalyptus Systems (2014a)
[171] Vgl. Eucalyptus Systems (2014c)
[172] Vgl. Hill, Z. (2013)
[173] Vgl. GitHub (2015e)

Ein Kalender und eine Kontaktdatenverwaltung werden durch Eucalyptus nicht zur Verfügung gestellt und jeweils mit null Punkten bewertet.

Für Eucalyptus existiert ein Authentifizierungsmodul.[175] Es konnten allerdings keine Anhaltspunkte für eine vorhandene 2FA gefunden werden. Daher wird für das Kriterium Authentifizierung ein Punkt vergeben.

Nicht-funktionale Kriterien:

Für das Kriterium Releaseabstände gibt es keine feste Angabe. Aus diesem Grund werden alle Releases seit Januar 2012[176]ausgewertet und ein Durchschnittswert ermittelt. Es werden hierbei keine vorläufigen Versionen oder Releases für veraltete Versionen, welche noch weiter supportet werden, beachtet. Die vollständige Übersicht ist dem Anhang 2 zu entnehmen. Mit einem durchschnittlichen Releaseabstand von 57,54 Tagen ist Eucalyptus im mittleren Bereich der untersuchten Produkte und erhält daher einen Punkt.

Es existiert ein vollständiges Benutzerhandbuch.[177] Soweit einschätzbar besitzt die Dokumentation eine gute Qualität und es wird die Punktzahl zwei vergeben. Es muss allerdings beachtet werden, dass etwaige Mängel in der Qualität der Dokumentation häufig erst bei einer Implementierung offensichtlich werden.

Das letzte Release von Eucalyptus wurde Ende Oktober veröffentlicht.[178] Zum Zeitpunkt des Abrufes ist zwischen einem und fünf Monaten her und das Kriterium Letztes Release somit mit der Punktzahl eins zu bewerten.

Die Dokumentation ist ausschließlich auf Englisch verfügbar.[179] Es wird bei Sprache der Dokumentation eine Punktzahl von eins vergeben.

Auf der offiziellen Seite von Eucalyptus sind Anfang Januar 2015 keine FAQ vorhanden.180 Deshalb werden für dieses Kriterium null Punkte vergeben.

Die übrigen nicht-funktionalen Kriterien werden in den Testberichten und wissenschaftlichen Ausarbeitungen im Vergleich zu den anderen drei Lösungen mäßig beurteilt.[181] Der größten Kritikpunkt muss dabei bei beim Durchsatz hingenommen werden.[182] Ferner sind die Autoren der übereinstimmenden Meinung, dass die bloße Optik – das Look and Feel – des Produkts Eucalyptus subjektiv am negativsten zu beurteilen ist.

[174] Vgl. Baun, C. u. a. (2011), S. 96
[175] Vgl. Eucalyptus Systems (2015d)
[176] Vgl. GitHub (2015d)
[177] Vgl. Eucalyptus Systems (2014b)
[178] Vgl. ebenda
[179] Vgl. ebenda
[180] Vgl. Eucalyptus Systems (2015c)
[181] Vgl. AL-Mukhtar, M. M. A./Mardan, A. A. A. (2014), S. 109 ff.
[182] Vgl. ebenda, S. 109 ff.

Open Source Kriterien:

Der globale Traffic Rank für eucalyptus.com beträgt Anfang Januar 349.297.[183] Dies ist der schlechteste Traffic Rank aller gemessenen Tools. Eucalyptus erhält daher hierfür null Punkte.

Das Projekt wurde im Herbst 2007 gestartet[184] und hat somit ein Alter von sieben Jahren. Dies entspricht zwei Punkten.

Für das Jahr 2014 wurden insgesamt 1131 Fehler dokumentiert.[185] Dies entspricht einer durchschnittlichen Fehlerrate von 94,25 Fehlern / Monat. Mit einer Fehlerrate zwischen 50 und 1.000 Fehlern/Monat ist Eucalyptus für dieses Kriterium die Punktezahl eins zuzuordnen.

Das Projekt wird von über zehn Stakeholdern unterstützt.[186] Dies entspricht einer Punktezahl von zwei.

Auf dem offiziellen Internetauftritt von Eucalyptus wird auf kein Forum verwiesen.[187] Daher ist der Aspekt Anzahl der Forumtopics mit 0 Punkten zu bewerten. Benutzer sind auf diese Weise gezwungen, sich selbst ein Forum auszusuchen und finden nicht den Großteil der Informationen gebündelt in einem Forum.

Auf der Internetseite von Eucalyptus ist neben einem Kontaktformular auch eine Telefonnummer angegeben.[188] Das Kriterium Direkter Ansprechpartner wird daher mit zwei Punkten bewertet.

Es gibt mehr als einen externen Provider, der ein Unternehmen entgeltlich bei der Installation einer Privat Cloud unterstützen kann.[189] Dies entspricht einer Wertung von zwei Punkten.

Im Januar 2015 sind auf Github 48 Contributers verzeichnet.[190] Hierfür werden null Punkte vergeben.

Bewertung des Tools CloudStack

Funktionale Kriterien:

Für CloudStack wird ein Hypervisor benötigt. Es werden KVM, Xen und vSphere unterstützt.[191] Für Plattform wird ein Punkt vergeben.

[183] Vgl. Alexa Internet (2015a)
[184] Vgl. Wolters, F. u. a. (2009)
[185] Vgl. Eucalyptus Systems (o. J.)
[186] Vgl. Eucalyptus Systems (2015e); vgl. dazu auch:Eucalyptus Systems (2015g)
[187] Vgl. Eucalyptus Systems (2015c)
[188] Vgl. Eucalyptus Systems (2015b)
[189] Vgl. Eucalyptus Systems (2015a); vgl. dazu auch: Eucalyptus Systems (2015f)
[190] Vgl. GitHub (2015c)
[191] Vgl. Baun, C. u. a. (2011), S. 102

CloudStack ist modular aufgebaut[192] und wird hierdurch mit zwei Punkten bewertet.

Monitoring wird für CloudStack über die VM Sync technology[193] zur Verfügung gestellt und mit zwei Punkten bewertet.

Für CloudStack ist ebenfalls eine Weboberfläche vorhanden.[194] Dies entspricht zwei Punkten.

Objektspeicher[195], Blockspeicher[196] und ein Netzwerkdienst[197] wird zur Verfügung gestellt. Dies wird jeweils mit zwei Punkten bewertet.

Eine Datenbank, ein Kalender und eine Kontaktdatenverwaltung wird nicht durch CloudStack zur Verfügung gestellt und daher mit null Punkten bewertet.

Es wird zudem eine eine Ein-Faktor-Authentifizierung ermöglicht. Dies entspricht einem Punkt.

<u>Nicht-funktionale Kriterien:</u>

Für CloudStack wird alle vier Monate ein neues Release veröffentlicht.[198] Dies ist im Bereich der Releaseabstände das Tool mit der niedrigsten Frequenz und somit mit null Punkten zu bewerten.

Das vollständige Benutzerhandbuch[199] besitzt, soweit abschätzbar, eine gute Qualität und wird mit der Punktzahl zwei bewertet.

Das letzte Release für die aktuelle Version von CloudStack würde Ende November veröffentlicht.[200] Dies liegt zum Zeitpunkt des Abrufes knapp über einen Monat zurück und wird mit der Punktzahl eins bewertet.

Die Dokumentation ist ausschließlich auf Englisch verfügbar[201] und wird mit einem Punkt bewertet.

Auf der offiziellen Seite von CloudStack sind FAQ vorhanden.[202] Dies entspricht zwei Punkten.

[192] Vgl. Shepherd, D. (2013)
[193] Vgl. Apache Software Foundation (2014b)
[194] Vgl. Büst, R. (2012)
[195] Vgl. Kannan, H. (2013)
[196] Vgl. Apache Software Foundation (2014b)
[197] Vgl. ebenda
[198] Vgl. Huang, A./Childers, C./Kinsella, J. (2013)
[199] Vgl. Apache Software Foundation (2014a)
[200] Vgl. GitHub (2015a)
[201] Vgl. Apache Software Foundation (2014f)
[202] Vgl. Apache Software Foundation (2014c)

Cloud Stack ist bezüglich der übrigen nicht-funktionalen Kriterien eine durchschnittliche Gewichtung zu geben. Positiv hebt sich hierbei die Sicherheit hervor, Kritikpunkte müssen bei den Möglichkeiten der Performance hingenommen werden.[203]

Open Source Kriterien:

Der globale Traffic Rank der offiziellen CloudStack-Seite http://cloudstack.apache.org/ ist nicht ermittelbar, da hierbei automatisch die übergeordnete Adresse apache.org ausgewertet wird. Diese Angabe besitzt keine Aussagekraft für das Tool CloudStack. Daher wird für CloudStack dieses Kriterium nicht bewertet und die Gewichtung hierbei für das Tool auf 0 gesetzt.

Das Tool existiert seit 2010[204] und hat somit ein Alter von vier Jahren. Dies entspricht einem Punkt.

Im Bug Tracking Tool von CloudStack wird für das Jahr 2014 nur ein Fehler angegeben.[205] Hiermit liegt die monatliche Fehlerrate bei < 1. Es sei an dieser Stelle zu erwähnen, dass die Angabe ungemein gering ist und ihre Korrektheit daher in Frage gestellt werden muss. Da es sich allerdings um den offiziellen Bug Tracker handelt, muss diese Information an dieser Stelle als korrekt angenommen werden. Das Kriterium wird daher mit zwei Punkten bewertet.

Das Projekt wird von über zehn Stakeholdern unterstützt.[206] Dies entspricht einer Punktezahl von zwei.

Ähnlich wie bei Eucalyptus verweist der Internetauftritt von CloudStack auf kein Forum. Das Kriterium Anzahl der Forumbeiträge wird daher mit null Punkten bewertet.

In Bezug auf direkte Ansprechpartner erhält man auf der Seite von CloudStack eine Liste mit vollständigen Namen.[207] Diese bietet allerdings keine einfache Möglichkeit einer Kontaktaufnahme. Daher wird dieses Kriterium mit null Punkten bewertet.

Für das Tool CloudStack existiert mehr als ein externer Provider[208]. Dies wird mit zwei Punkten bewertet.

Auf der Seite Github sind im Januar 2015 für CloudStack 171 Contributers verzeichnet.[209] Dies entspricht einem Punkt.

[203] Vgl. AL-Mukhtar, M. M. A./Mardan, A. A. A. (2014), S. 109 ff.
[204] Vgl. Apache Software Foundation (2014e)
[205] Vgl. Apache Software Foundation (o. J.)
[206] Vgl. Apache Software Foundation (2012)
[207] Vgl. Apache Software Foundation (2014d)
[208] Vgl. ActOnMagic Technologies (2014); vgl. dazu auch: Appcore (2014)
[209] Vgl. GitHub (2015b)

Bewertung des Tools OpenStack

Funktionale Kriterien:

Bei OpenStack wird ein Hypervisor benötigt. Es wird KVM, Xen, VirtualBox und UML unterstützt.[210] Dies entspricht für einem Punkt für das Kriterium Plattform.

In Abb. 3 wird ersichtlich, dass OpenStack modular aufgebaut ist. Das Tool erhält daher für das Kriterium Modularität zwei Punkte.

Das Modul Neutron ermöglicht Montoring[211] hierfür erhält OpenStack zwei Punkte.

Auch für OpenStack ist eine Weboberfläche vorhanden.[212] Dies entspricht zwei Punkten.

Es werden ein Objektspeicher[213], ein Blockspeicher[214], eine Datenbank[215] und ein Netzwerkdienst[216] zur Verfügung gestellt. Hierfür werden jeweils zwei Punkte vergeben.

Es wird ein Kalender noch eine Kontaktdatenverwaltung durch OpenStack zur Verfügung gestellt. Hierfür werden jeweils null Punkte vergeben.

2FA wird durch OpenStack unterstützt.[217] Hierfür werden zwei Punkte vergeben.

Nicht-funktionale Kriterien:

Im Bereich der Releaseabstände existiert für das Tool keine feste Frequenz. Es wird daher der durchschnittliche Abstand aller Releases ab der Serie Essex ermittelt.[218] Hierbei finden Releases für alte Serien, welche weiter supportet werden, keine Beachtung. Die vollständige Auswertung ist dem Anhang 3 zu entnehmen. Zwischen den Releases liegt durchschnittlich ein Abstand von 46,38 Tagen. Damit befindet sich das Tool OpenStack im mittleren Bereich und erhält einen Punkt.

Das Benutzerhandbuch ist vollständig und hat eine gute Qualität zu besitzen.[219] Dies wird mit zwei Punkten bewertet.

Oktober 2014 wurde das letzte Release für OpenStack veröffentlicht.[220] Zum Zeitpunkt des Abrufes ist dies zwischen einem und fünf Monaten her und entspricht somit einem Punkt.

[210] Vgl. Baun, C. u. a. (2011), S. 101
[211] Vgl. OpenStack (o. J.e)
[212] Vgl. Büst, R. (2012)
[213] Vgl. Baun, C. u. a. (2011), S. 102
[214] Vgl. OpenStack (o. J.j)
[215] Vgl. OpenStack (o. J.j)
[216] Vgl. OpenStack (o. J.i)
[217] Vgl. OpenStack (o. J.a)
[218] Vgl. OpenStack (o. J.l)
[219] Vgl. OpenStack (o. J.c)
[220] Vgl. OpenStack (o. J.k)

Die Dokumentation ist Ende Dezember 2014 nur sehr begrenzt auf Deutsch verfügbar.[221] Dies wird als nicht ausreichend für eine Vergabe von zwei Punkten bewertet. Das Tool OpenStack erhält für das Kriterium Sprache der Dokumentation einen Punkt.

Auf der offiziellen Seite von OpenStack sind FAQ vorhanden.[222] Dies entspricht zwei Punkten.

Die restlichen nicht-funktionalen Bewertungskriterien von OpenStack sind durchschnittlich einzustufen. Es wird für die verbleibenden Faktoren die mittlere Bewertungsstufe mit einem Punkt vergeben, da keine überragenden Eigenschaften zu finden sind.[223] Andererseits muss hierbei auch festgehalten werden, dass es keine nennenswerten negativen Kritiken gab[224], weshalb die Vergabe von null Punkten nicht gerechtfertigt ist.

Open Source Kriterien:

Der globale Traffic Rank von OpenStack ist Anfang 2015 17.323.[225] Dieser Traffic Rank ist weder der beste noch der schlechteste Traffic Rank und erhält somit einen Punkt.

Das Tool existiert seit vier Jahren.[226] Dies entspricht einem Punkt.

Für die Fehlerrate werden nicht alle Module berücksichtigt. Für eine funktionierende Implementierung werden folgende Module als notwendig betrachtet und entsprechend ausgewertet: Nova, Swift, Cinder, Neutron und Horizon. Im Bug Tracking Tool von OpenStack muss jedes Modul einzeln ausgewertet werden. Diese Auswertung muss manuell erfordern, da keine Selektion nach Erstellungsdatum erfolgen kann. Dies ist sehr fehleranfällig. Zudem fällt auf, dass auch die Sortierung nach „newest first“ nicht immer korrekt funktioniert. Für Nova können 1654 Fehler in 2014 ermittelt werden,[227] für Swift 207[228] und für Cinder 881 Fehler[229]. Für Neutron wurden im Jahre 2014 1194 Fehler gemeldet[230] und für Horizon 1194[231]. Hieraus ergeben sich 5554 Fehler für 2014. Dies entspricht einer monatlichen Fehlerrate von 462,83 und einer Punktzahl von eins.

Das Projekt OpenStack hat weit über zehn Stackholder[232] und erhält daher zwei Punkte.

[221] Vgl. OpenStack (o. J.d)
[222] Vgl. OpenStack (o. J.h)
[223] Vgl. Burns, C. (2013); vgl. dazu auch: Steinmetz, D. u. a. (2012)
[224] Vgl. Burns, C. (2013)
[225] Vgl. Alexa Internet (2015b)
[226] Vgl. The Register (2010)
[227] Vgl. Canonical (2014b)
[228] Vgl. Canonical (2014e)
[229] Vgl. Canonical (2014a)
[230] Vgl. Canonical (2014d)
[231] Vgl. Canonical (2014c)
[232] Vgl. OpenStack (o. J.b)

Das Forum enthält zum Zeitpunkt des Abrufes 10.694 Topics.[233] Dies entspricht einer Punktzahl von eins.

Für OpenStack existieren keine direkte Ansprechpartner, sondern nur sogenannte Mailinglisten.[234] Dieses Kriterium wird daher mit null Punkten bewertet.

Es existiert mehr als ein externer Provider für das Tool OpenStack.[235] Hierfür werden zwei Punkte vergeben.

Für OpenStack sind im Januar 2015 412 Contributers vermerkt.[236] Dies wird mit zwei Punkten bewertet.

Bewertung des Tools ownCloud

Funktionale Kriterien:

Im Gegensatz zu den restlichen untersuchten Tools benötigt ownCloud keinen Hypervisor. Eine Windows- oder Linux-System ist ausreichend.[237] Aus diesem Grund werden hierfür zwei Punkte für das Kriterium Plattform vergeben.

Bei ownCloud ist die Modularität gegeben, da das Produkt über Apps erweiterbar ist.[238] Dies entspricht zwei Punkten.

Bei ownCloud ist Monitoring über eine App möglich.[239] Dies entspricht zwei Punkten.

ownCloud stellt eine Weboberfläche zur Verfügung.[240] Dies entspricht einer Punktzahl von zwei.

ownCloud stellt weder einen Objekt- noch einen Blockspeicher zur Verfügung. Zudem werden auch keine Datenbank und kein Netzwerkdienst angeboten. Für diese Kriterien werden jeweils null Punkte vergeben.

Das Tool ownCloud stellt einen Kalender und eine Kontaktdatenverwaltung zur Verfügung.[241] Dies wird mit jeweils zwei Punkten bewertet.

Bei ownCloud wird eine 2FA über eine externe App ermöglicht.[242] Dies entspricht zwei Punkten.

[233] Vgl. OpenStack (o. J.g)
[234] Vgl. OpenStack (o. J.f)
[235] Vgl. Rackspace (o. J.); vgl. dazu auch: Host Europe (o. J.)
[236] Vgl. GitHub (2015f)
[237] Vgl. ownCloud (2015c)
[238] Vgl. ownCloud (o. J.d)
[239] Vgl. Paysant, P. (2014)
[240] Vgl. ownCloud (o. J.a)
[241] Vgl. Porteck, S. (2013)
[242] Vgl. Gude, S. (2014)

Nicht-funktionale Kriterien:

Für das Tool ownCloud ist keine feste Releasefrequenz festgelegt. Die Auswertung der Releases ab der Version 3.0.1[243] ergibt einen durchschnittlichen Abstand von 23,33 Tagen zwischen zwei Releases. Die vollständige Auswertung ist dem Anhang 4 zu entnehmen. Releases für alte Versionen wurden abermals nicht beachtet. ownCloud besitzt den geringsten Releaseabstand und erhält hierfür zwei Punkte.

Es existiert ein vollständiges Benutzerhandbuch.[244] Soweit einschätzbar besitzt die Dokumentation eine gute Qualität und es wird die Punktzahl zwei vergeben. Dies stimmt auch mit den in Kapitel 0 gemachten Erfahrungen überein.

Das letzte Release von ownCloud wurde im Dezember 2014 veröffentlicht.[245] Zum Zeitpunkt des Abrufes ist dies weniger als einen Monat her. Dies ist somit der Punktzahl zwei zu bewerten.

Die Dokumentation ist ausschließlich auf Englisch verfügbar.[246] Hierfür wird eine Punktzahl von eins vergeben.

Auf der offiziellen Seite von ownCloud sind sind FAQ vorhanden.[247] Deshalb werden für dieses Kriterium zwei Punkte vergeben.

Die übrigen nicht-funktionalen Kriterien des Cloud-Produkts ownCloud dürfen vorwiegend positiv bewertet werden. Es wird im direkten Vergleich mit den Alternativprodukten in wissenschaftlichen Texten positiv bewertet. Negative Punkte, vor allem was Kapazität und Performance betrifft, sind im Gegensatz zu den anderen drei Lösungen, nicht veröffentlicht.[248]

Open Source Kriterien:

Der globale Traffic Rank für owncloud.org beträgt Anfang Januar 15.901.[249] Dies ist der höchste Traffic Rank aller untersuchten Tools. Dies entspricht zwei Punkten.

ownCloud existiert seit 2010 und ist somit vier Jahre alt.[250] Dies entspricht einer Punktzahl von eins.

[243] Vgl. o. V. (o. J.a)
[244] Vgl. ownCloud (o. J.b)
[245] Vgl. ownCloud (2015e)
[246] Vgl. ownCloud (o. J.b)
[247] Vgl. ownCloud (2015b)
[248] Vgl. Burns, C. (2013)
[249] Vgl. Alexa Internet (2014)
[250] Vgl. Karlitschek, F. (2010)

Für das Jahr 2014 wurden insgesamt 2.487 Fehler dokumentiert.[251] Dies entspricht einer durchschnittlichen Fehlerrate von 207,25 Fehlern / Monat. Mit einer Fehlerrate zwischen 50 und 1.000 Fehlern / Monat wird ownCloud ein Punkt zugeordnet.

Für die Community Cloud von ownCloud können keine Stakeholder ermittelt werden. Diese werden nur für die Enterprise Edition aufgeführt.[252] Dieses Kriterium wird mit null Punkten bewertet.

Das Forum von ownCloud hat zum Zeitpunkt des Aufrufes 15.142 Topics.[253] Dies entspricht einer Bewertung von einem Punkt.

Für ownCloud gibt es eine Übersicht mit direkten Ansprechpartnern mit Kontaktmöglichkeiten über verschiedene Social Media Plattformen.[254] Dies wird mit zwei Punkten bewertet.

Es konnte trotz längerer Suche nur ein externer Provider für die Community Edition von ownCloud gefunden werden.[255] Hierfür wird ein Punkt vergeben.

Auf Github sind bei ownCloud im Januar 2015 285 Contributers verzeichnet.[256] Hierfür werden zwei Punkte vergeben.

Übersicht

Nach der vollständigen Vergabe aller Punkte ergibt sich folgende Übersicht:

8 Anwendung des Kriterienkatalogs

Autoren: Amelie Böer, Thomas Enderle, Matthias Holzwarth, Dominik Schrahe

Im folgenden Kapitel wird der erstellte Kriterienkatalog angewandt. Hierbei wird die Gewichtung durchgeführt und das Ergebnis berechnet. Im Rahmen dieser Arbeit wird dieser Schritt zweimal durchgeführt. Zu Beginn wird die Gewichtung für die _Versicherung (Auftraggeber) in Zusammenarbeit mit einem Ansprechpartner im Unternehmen durchgeführt. Anschließend wird dieser Schritt durch das Forschungsteam selbst für ein fiktives kleineres Unternehmen erneut durchgeführt. Ziel dieser doppelten Anwendung ist zu untersuchen, in wie weit sich die Ergebnisse für verschiedene Unternehmenstypen voneinander unterscheiden. Zur Vereinfachung der Durchführung wird zur Auswertung ein einfaches Tool mit Hilfe von Excel entwickelt, welches das Ergebnis automatisiert nach Eingabe der gewählten Gewichtung auf

[251] Vgl. GitHub (2015g)
[252] Vgl. ownCloud (2015d)
[253] Vgl. ownCloud (o. J.e)
[254] Vgl. ownCloud (2015a)
[255] Vgl. Kangaroot (2015)
[256] Vgl. GitHub (2015h)

Grundlage der vergebenen Punkte berechnet. Dieses wird zu Beginn des Kapitels knapp dargestellt.

Anleitung zur Benutzung des Tools

Zur leichteren Auswertung des Ergebnisses wird ein einfaches Tool mittels Excel entwickelt. Hierbei ist zu beachten, dass die Erstellung des Tools kein zentrales Ziel dieser Arbeit ist und somit die Grundsätze der Softwareergonomie nur eingeschränkt Anwendung finden. Erweiterungsmöglichkeiten, z.B. um Barrierefreiheiten, werden am Ende dieses Kapitels erläutert.

Im Anschluss befindet sich eine kurze Anleitung zur Benutzung des Tools. Der genaue Aufbau bzw. weitere Details zur Erstellung des Tools werden in Anhang 5 aufgezeigt.

Das Tabellenblatt Übersicht, dargestellt in Abb. 11 ist die Hauptansicht des Tools. Hier ist die Eintragung der gewünschten Gewichtung und die hieraus resultierenden Empfehlung ersichtlich. Anschließend folgen die Tabellenblätter mit den detaillierten Auswertungen für die einzelnen Tools. Im letzten Arbeitsblatt befindet sich die Bewertungsgrundlage mit einer Übersicht der jeweiligen Bewertungen der Produkte für alle Kriterien.

Vor der Eintragung der Gewichtung sollte der Kriterienkatalog als Nachschlagewerk vorliegend sein. Zusätzlich wird empfohlen, sowohl die grafische Übersicht zur Gewichtungsempfehlung als auch die ausführliche schriftliche Erläuterung zur Verfügung zu haben. Alle graumarkierten Felder sind auszufüllen. Alle grau geschriebenen Kriterien werden mitbeachtet, hier ist allerdings keine Gewichtung nötig bzw. möglich.

Die Auswertung erfolgt in drei Schritten.

Der erste Schritt ist die Eintragung der unternehmensspezifischen Gewichtung. Auf der Übersicht gibt es drei farbig markierte Bereiche. Der blaue obere Bereich ist zuständig für alle funktionalen Kriterien. Hier ist eine Eintragung ab null möglich. Für diesen Bereich ist keine Gewichtungsempfehlung möglich, da dies allein abhängig vom gewünschten Einsatz ist. Für eine genauere Erläuterung der Kriterien werden im Kriterienkatalog Hilfestellungen gegeben. Alle für die Anwendung der Cloud nicht benötigten oder nicht geplanten Funktionen sollten dringendst mit null gewichtet werden. Alle restlichen Kriterien sind prinzipiell offen und sollten nach der Wichtigkeit bewertet werden. An dieser Stelle wird zunächst eine Gewichtung bis zu fünf empfohlen. Der grüne mittlere Bereich entspricht den nicht-funktionalen Kriterien, der gelbe untere Bereich den Open Source-spezifischen Kriterien. Diese Bereiche sind gleich zu behandeln und es gibt eine entsprechende Empfehlung. Diese ist allerdings nicht verpflichtend. Es sind Gewichtungen ab eins erlaubt. Prinzipiell ist die Gewichtung abermals nach oben offen, es wird allerdings eine Einschränkung bis fünf empfohlen.

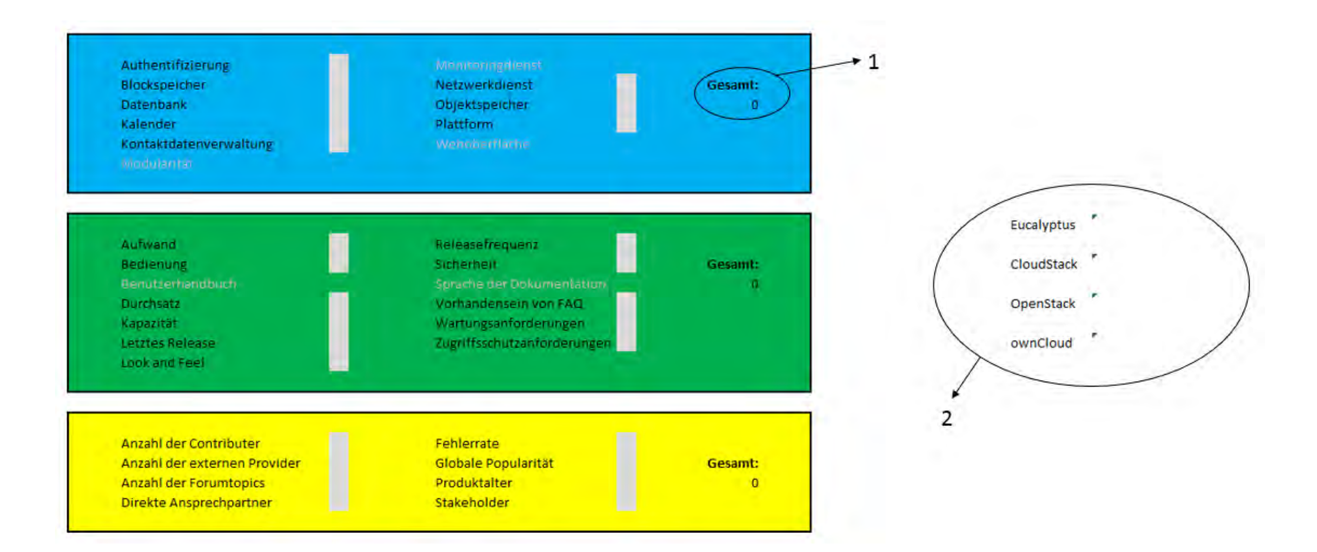


Abb. 11: Auswertungstool Hauptansicht

Nach der Eintragung aller unternehmensspezifischen Gewichtungen muss nun im nächsten Schritt die Gewichtung der einzelnen Bereiche betrachtet werden. Es sollte sichergestellt werden, dass der Bereich der funktionalen Kriterien nicht zu gering gewichtet wird. Hierzu bitte die Gesamt-Angaben, vgl. Abb. 11 Ziffer 1, betrachten. Auch das beste Produkt ist für ein Unternehmen nutzlos, wenn es nicht die geforderten Funktionen bereitstellt. Hieraus besteht die Notwendigkeit, dass der Bereich der funktionalen Kriterien nicht geringer sein darf als einer der beiden anderen Bereiche. Falls dies der Fall ist, bitte alle funktionalen Kriterien um einen entsprechenden Faktor gleichmäßig erhöhen. Dies gilt auch, wenn dadurch die Gewichtung fünf übersteigt. Anschließend ist das Ergebnis rechts, vgl. Abb. 11, Ziffer 2, abzulesen. Die Prozentzahl drückt hierbei den Erreichungsgrad aus, das Produkt mit der höchsten Prozentzahl wird empfohlen.

Eucalyptus					
	Kriterium	Punktzahl	Gewichtung	Kriterien-punktzahl	Maximal zu erreichende Punktzahl
Funktionale Kriterien	Authentifizierung	1	0	0	0
	Blockspeicher	2	0	0	0
	Datenbank	2	0	0	0
	Kalender	0	0	0	0
	Kontaktdatenverwaltung	0	0	0	0
	Modularität	2	0	0	0
	Monitoringdienst	2	0	0	0
	Netzwerkdienst	2	0	0	0
	Objektspeicher	2	0	0	0
	Plattform	1	0	0	0
	Weboberfläche	2	0	0	0
Nicht-funktionale Kriterien	Aufwand	1	0	0	0
	Bedienung	1	0	0	0
	Benutzerhandbuch	2	0	0	0
	Durchsatz	0	0	0	0
	Kapazität	1	0	0	0
	Letztes Release	1	0	0	0
	Look and Feel	0	0	0	0
	Releasefrequenz	1	0	0	0
	Sicherheit	0	0	0	0
	Sprache der Dokumentation	1	0	0	0
	Vorhandensein von FAQ	0	0	0	0
	Wartungsanforderungen	1	0	0	0
	Zugriffsschutzanforderungen	2	0	0	0
Open Source Kriterien	Anzahl der Contributer	0	0	0	0
	Anzahl der Externen Provider	2	0	0	0
	Anzahl der Forumtopics	0	0	0	0
	Direkte Ansprechpartner	2	0	0	0
	Fehlerrate	1	0	0	0
	Globale Populariät (Traffic Rank)	0	0	0	0
	Produktalter	0	0	0	0
	Stakeholder	2	0	0	0

Erreichte Punktzahl	Maximal erreichbare Punktzahl	Ergebnis
0	0	

Abb. 12: Auswertungstool Detailansicht Eucalyptus

Der dritte Schritt, die Sichtung der detaillierten Ergebnisse, ist optional. Es wird allerdings dringend empfohlen, zumindest das gewählte Produkt zu überprüfen. Hierfür wird auf das Tabellenblatt des jeweiligen Tools geklickt. Anschließend wird die Übersicht, vgl. Abb. 12 mit Eucalyptus, angezeigt. Rechts, vgl. Abb. 12 Ziffer 1, befindet sich eine Übersicht aller Kriterien. Horizontal, s. Abb. 12 Ziffer 2, ist nun die erreichte Punktzahl der Cloud im entsprechenden Kriterium eingetragen. Anschließend ist die vom Unternehmen gewählte Gewichtung für das Kriterium ersichtlich. Aus der Punktzahl und der Gewichtung ergibt sich die erreichte Kriterienpunktzahl der Cloud für dieses eine Kriterium. Rechts hiervon ist die in diesem Kriterium maximal mögliche Punktzahl dargestellt. Alle Bereiche mit einer Gewichtung größer vier werden zur besseren Übersicht gelb markiert. Im unteren Bereich, vgl. Abb. 12 Ziffer 3, ist die erreichte Gesamtpunktzahl der Cloud und die maximal erreichbare Gesamtpunktzahl einander gegenübergestellt. Hieraus errechnet sich die prozentuale Zielerreichung, vgl. Abb. 12 Ziffer 4. Auf dieser Übersicht sollte nun kontrolliert werden, in wie weit alle gelb markierten Kriterien zufriedenstellen erfüllt sind. Auch wenn eine Cloud eine vergleichsweise hohe Prozentzahl erreicht, kann es vorkommen, dass ein für den Einsatz not-

wendiges Kriterium nicht erfüllt ist und dadurch eine andere Cloud vorteilhafter ist. In diesem Fall sollte das nächsthohe Produkt untersucht werden.

Gewichtungsempfehlung

Im ersten Bereich, den funktionalen Kriterien, ist eine Gewichtung beginnend bei null zulässig. Hierbei ist zu beachten, dass alle Funktionen, welche aktuell nicht benötigt und in naher Zukunft auch nicht geplant werden, mit Null zu gewichten sind. Zudem ist nach der Durchführung bei der Detailansicht ein besonderes Augenmerk darauf zu legen, ob alle dringend benötigten Funktionen auch tatsächlich vorhanden sind.

In den Bereichen der nicht-funktionalen und der Open Source Kriterien ist eine Gewichtung ab eins zulässig. Im Folgenden wird eine Empfehlung für eine Gewichtung gegeben. Es ist hierbei zu beachten, dass diese keine allgemeine Gültigkeit besitzt, sondern sich abhängig vom Unternehmen und geplanten Einsatzzweck auch ändern kann. Die empfohlene Gewichtung sollte daher nicht einfach blind übernommen werden, sondern auf die Gültigkeit im vorliegenden Fall geprüft werden. In Anhang 6 befindet sich eine Übersicht der empfohlenen Gewichtung für eine einfache Anwendbarkeit.

Die beiden Kriterien Releasefrequenz und Letztes Release spiegeln beide Sicherheitsaspekte wieder. Da das letzte Release allerdings eine stichtagsbezogene Betrachtung ist, besitzt es eine geringere Aussagekraft. Es wird eine Gewichtung von vier für Releasefrequenz und eins für letztes Release empfohlen. Das Kriterium Sicherheit ist allgemeiner gewählt und sollte mit fünf gewichtet werden.

Das Vorhandensein von FAQ soll, genau wie das Benutzerhandbuch und die Sprache der Dokumentation, dem Benutzer die Anwendung erleichtern. Das Kriterium sollte für gewöhnlich mit zwei Punkten gewichtet werden. Haben Mitarbeiter bereits Erfahrung mit den jeweiligen Tools, kann das Kriterium allerdings auf eins gesenkt werden.

Das Look and Feel trägt dazu bei, dass die Mitarbeiter leichter von der Einführung des Tools überzeugt werden. Zudem arbeiten die Mitarbeiter bei einem Tool mit gutem Look and Feel gerne und häufig produktiver. Das Kriterium wird als sehr wichtig angesehen und sollte mit vier gewichtet werden. Wird die Einführung von den Mitarbeitern selbst vorangetrieben, kann die Gewichtung verringert werden. Sollen in erster Linie IT-fremde Mitarbeiter mit dem Tool arbeiten, sollte die Gewichtung erhöht werden.

Die Kriterien Durchsatz und Kapazität sind allein von der geplanten Anwendung und der benötigten Datenmenge abhängig. Hierbei kann keine gültige Empfehlung ausgesprochen werden. Dies muss daher individuell für den jeweiligen Anwendungsfall entschieden werden.

Der Aufwand ist abhängig vom jeweiligen Unternehmen. Bei einem großen oder mittleren Unternehmen mit eigener IT-Abteilung ist auch ein größerer Aufwand machbar, da sich

dieser auf mehrere Personen verteilt. Hier wird eine Gewichtung zwischen eins und drei empfohlen. Für eher kleinere Unternehmen ist ein sehr großer Aufwand allerdings nicht möglich. Hier sollte eine Gewichtung von vier oder fünf erfolgen.

Die Bedienung ist allgemein ein sehr wichtiger Faktor. Eine selbsterklärende Bedienung bedeutet zudem ein geringerer Aufwand an Schulungen / Schulungsunterlagen. Es sollte eine Gewichtung zwischen vier und fünf gewählt werden.

Mit den Wartungsanforderungen verhält es sich ähnlich wie mit dem Aufwand. Empfohlen wird daher eine Gewichtung um eins höher als der Aufwand. Auslöser hierfür ist, dass der Aufwand nur einmalig anfällt. Die Wartung allerdings einen fortlaufenden Faktor darstellt.

Die Zugriffsschutzanforderungen sind ein Sicherheitskriterium. Je nach Sensibilität der gespeicherten Daten muss die Gewichtung entsprechend höher gelegt werden. Für Versicherungsunternehmen wird allgemein eine Gewichtung zwischen vier und 5 empfohlen.

Bei einer hohen globalen Popularität ist davon auszugehen, dass aktuell ein großes Interesse für das Produkt besteht. Es gibt allerdings keine Aussage darüber, ob es sich nur um einen kurzfristigen Trend handelt. Es ist daher weniger aussagekräftig als das Produktalter und sollte mit zwei gewichtet werden.

Bei sehr jungen Produkten herrscht Unsicherheit, ob es sich bei diesem nur um einen kurzfristigen Trend handelt. Daher sollte eine Gewichtung von drei gewählt werden.

Die Fehlerrate ist, wie bereits aufgeführt, nur eingeschränkt aussagekräftig, da eine hohe Fehlerrate in Ausnahmefällen nicht unbedingt negativ zu bewerten ist. Daher sollte dieses Kriterium mit eins gewichtet werden.

Das Vorhandensein von Stakeholdern ist besonders für kleine bis mittlere Firmen von Bedeutung, da die Stakeholder ein gewisses Interesse an einer Beständigkeit des Produktes haben. Dies ist ein Indiz dafür, dass es sich nicht um einen kurzfristigen Trend handelt. Hier sollte eine Gewichtung von vier erfolge. Große Firmen können hingegen selbst zu einem Stakeholder werden. Hier ist eine Gewichtung von zwei angemessen.

Die Anzahl der Forumtopics ist sehr wichtig, da hier schnell und sofort Hilfe gefunden werden kann. Es wird eine Gewichtung von vier empfohlen.

Direkte Ansprechpartner sind praktisch, allerdings bei weitem nicht so bedeutend wie die Anzahl der Forumtopics. Zum einen muss man hierbei ggf. auf die Antwort des Ansprechpartners länger warten. Zudem ist hier keine Antwort gewährleistet. Es wird eine Gewichtung von zwei empfohlen.

Bei einer hohen Anzahl von Contributern, wird das Produkt tendenziell schneller weiterentwickelt und Fehler können schneller gefunden / ausgebessert werden. Dies ist ein sehr wichtiges Kriterium und sollte mit vier gewichtet werden.

Das Vorhandensein von externen Providern ist nur wichtig, wenn diese auch genutzt werden sollen. Soll ein Provider gefunden werden, ist eine Gewichtung von fünf angemessen. Ist ein Provider gewünscht, aber optional, ist drei ausreichend. Ist kein Provider, weder gegenwärtig noch für die Zukunft geplant, sollte dieses Kriterium mit eins gewichtet werden.

Anwendung durch das Unternehmen

Das erstellte Tool wird dem kooperierenden Unternehmen zusammen mit dem Kriterienkatalog, der Anleitung und den Gewichtungsempfehlung übergeben.

Das Ergebnis nach der ersten Gewichtung ist in Abb. 13 abgebildet. In diesem Fall gewinnt ownCloud.

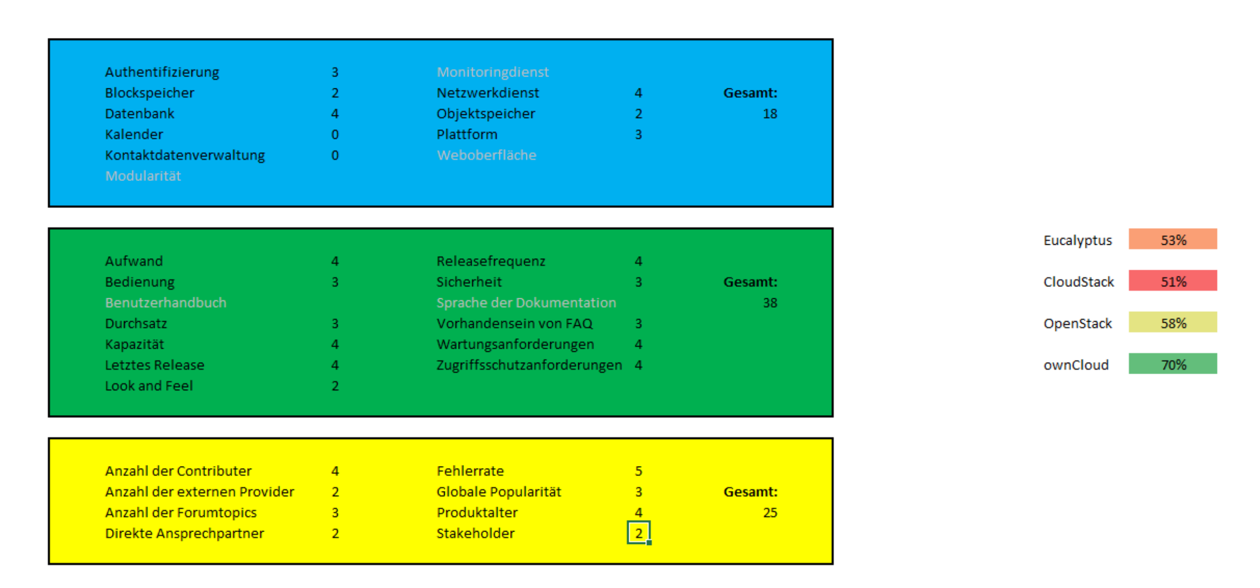


Abb. 13: _Versicherung – Gesamtübersicht nach dem ersten Gewichtungsschritt

Es fällt allerdings auf, dass die Gesamtpunktevergabe für die funktionalen Kriterien weit unter den anderen Bereichen liegt. Es ist also eine Erhöhung der funktionalen Kriterien notwendig, um ein ausgeglichenes Ergebnis zu erzielen. Untersucht man das Produkt ownCloud in der Detailansicht, wird die Notwendigkeit des Gewichtungsverhältnisses sehr deutlich. Ein Ausschnitt der endsprechenden Ansicht ist Abb. 14 zu entnehmen. Hier werden die gewünschten funktionalen Kriterien nicht erfüllt.

ownCloud					
	Kriterium	Punktzahl	Gewichtung	Kriterien-punktzahl	Maximal zu erreichende Punktzahl
Funktionale Kriterien	Authentifizierung	2	3	6	6
	Blockspeicher	0	2	0	4
	Datenbank	0	4	0	8
	Kalender	2	0	0	0
	Kontaktdatenverwaltung	2	0	0	0
	Modularität	2	0	0	0
	Monitoringdienst	2	0	0	0
	Netzwerkdienst	0	4	0	8
	Objektspeicher	0	2	0	4
	Plattform	2	3	6	6
	Weboberfläche	2	0	0	0
	Aufwand	2	4	8	8

Abb. 14: _Versicherung – Detailansicht ownCloud nach dem ersten Gewichtungsschritt

Es wird der Faktor drei gewählt. Hierdurch liegt die Gesamtgewichtung der funktionalen Kriterien über denen der anderen Bereiche. Der Abb. 15 ist das endgültige Ergebnis zu entnehmen.

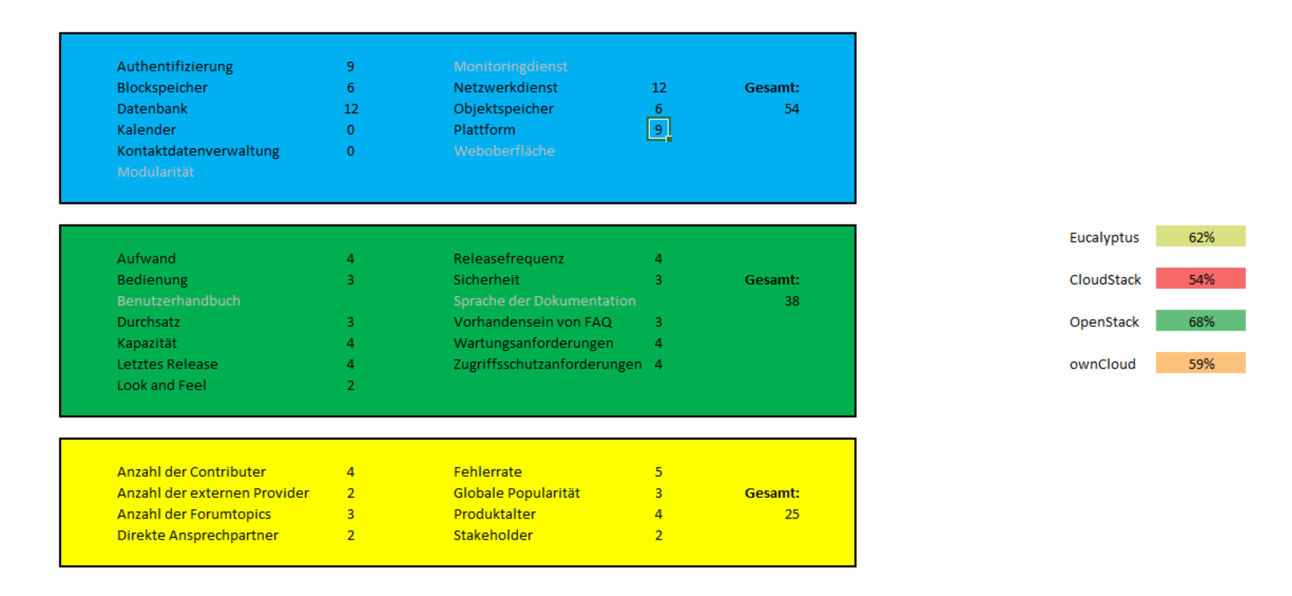


Abb. 15: _Versicherung – Gesamtübersicht nach dem zweiten Gewichtungsschritt

Das empfohlene Produkt ist OpenStack. In Abb. 16 ist ein Ausschnitt der Detailansicht von OpenStack zu sehen. Es fällt auf, dass im Gegensatz zu ownCloud hier die funktionalen Wünsche alle erfüllt sind.

OpenStack					
	Kriterium	Punktzahl	Gewichtung	Kriterien-punktzahl	Maximal zu erreichende Punktzahl
Funktionale Kriterien	Authentifizierung	2	9	18	18
	Blockspeicher	2	6	12	12
	Datenbank	2	12	24	24
	Kalender	0	0	0	0
	Kontaktdatenverwaltung	0	0	0	0
	Modularität	2	0	0	0
	Monitoringdienst	2	0	0	0
	Netzwerkdienst	2	12	24	24
	Objektspeicher	2	6	12	12
	Plattform	1	9	9	18
	Weboberfläche	2	0	0	0
	Aufwand	0	4	0	8

Abb. 16: _Versicherung – Detailansicht OpenStack nach dem zweiten Gewichtungsschritt

Anwendung durch das Projektteam

Versicherungs-Maklerbüro Freund ist ein sehr modernes Unternehmen mit aktuell 15 Mitarbeitern. Derzeit kümmert sich eine Person um die IT. Die Mitarbeiter arbeiten sehr viel vor Ort beim Kunden. Es herrscht eine „Bring your own device“-Policy. Das Arbeiten muss grundsätzlich mit allen Geräten und überall möglich sein. Aus diesem Grund zieht Freund die Anwendung einer Privat Cloud in Betracht. Die Zusammenarbeit zwischen Mitarbeitern und die Dateisynchronisation stehen zunächst im Vordergrund. Bei positiven Erfahrungen sollen dauerhaft auch die Kalender- und die Kontaktfunktionen über die Cloud organisiert werden.

Begründung der Gewichtung

Anbei wird die für Maklerbüro Freund gewählte Gewichtung erläutert. Die erstellte Gewichtungsempfehlung wird strikt angewendet.

Aufgrund der begrenzten IT-Ressourcen wird das Kriterium Plattform mit fünf gewichtet. Es wird mit keiner sehr großen Datenmenge gerechnet. Daher wird der Objektspeicher mit eins gewichtet. Das Maklerbüro benötigt keinen Blockspeicher, dieses Kriterium wird mit null gewichtet. Ebenso wird eine Datenbank aktuell nicht benötigt. Je nach Entwicklung des Kundenstamms könnte allerdings dauerhaft die Auslagerung und Verwaltung der Kundenkontaktdaten in eine Datenbank sinnvoll sein. Daher wird das Kriterium Datenbank mit eins gewichtet. Ein Netzwerkdienst ist nicht von zentraler Bedeutung und wird daher mit eins gewichtet. Die Kalenderfunktion und Kontaktdatenverwaltung ist aktuell nicht vorrangig von Bedeutung, soll allerdings dauerhaft möglich sein. Daher werden diese beiden Kriterien jeweils mit vier gewichtet. Eine möglichst sichere Authentifizierung ist Freund besonders wichtig. Es wird mit vier gewichtet. Der Aufwand wird mit fünf gewichtet, da nur eine Person für die IT zuständig ist. Da von einer sehr geringen Datenmenge ausgegangen wird, werden der Durchsatz und die Kapazität jeweils mit eins gewichtet. Das Look and Feel wird mit fünf gewichtet, da die IT-fremden Mitarbeiter von der Anwendung überzeugt werden müssen.

Das Vorhandensein von FAQ wird hingegen mit zwei gewichtet. Die Zugriffsschutzanforderungen werden mit fünf gewichtet, um der Bedeutung der Sicherheit Ausdruck zu geben. Die Anzahl der externen Provider wird mit drei gewichtet. Es soll nicht von Beginn an mit einem externen Provider zusammengearbeitet werden. Freund möchte sich die Option allerdings offenhalten, dies auf Dauer zu tun. Alle weiteren Gewichtungen werden wie in der Empfehlung durchgeführt.

Das Ergebnis des ersten Gewichtungsschritts ist in Abb. 17 dargestellt.

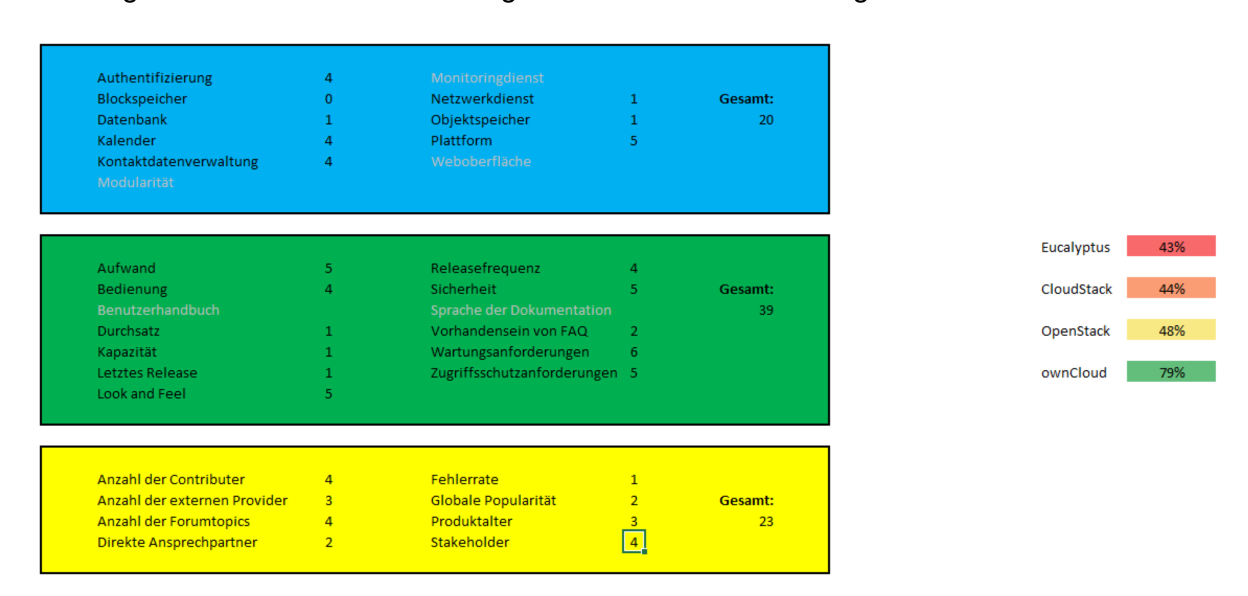


Abb. 17: Freund – Gesamtübersicht nach der ersten Gewichtungsempfehlung

Der funktionale Bereich ist niedriger als die anderen Bereiche und wird daher um den Faktor zwei erhöht. Das Ergebnis nach dem zweiten Gewichtungsschritt ist Abb. 18 zu entnehmen.

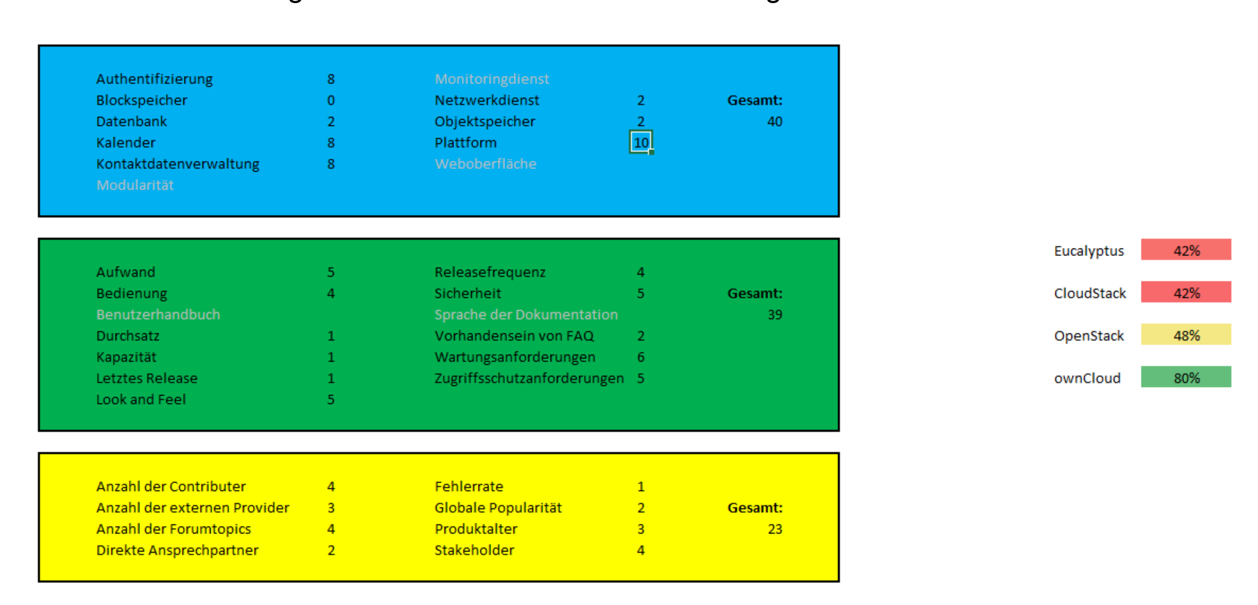


Abb. 18: Freund – Gesamtübersicht nach dem zweiten Gewichtungsschritt

Das empfohlene Produkt sollte nun noch untersucht werden. In Abb. 19 ist ein Ausschnitt aus der Detailansicht für ownCloud zu sehen. Alle Kriterien, welche mit mehr als vier gewichtet wurden, sind gelb markiert. Es fällt auf, dass alle benötigten Funktionen vorhanden sind.

ownCloud					
	Kriterium	Punktzahl	Gewichtung	Kriterien-punktzahl	Maximal zu erreichende Punktzahl
Funktionale Kriterien	Authentifizierung	2	8	16	16
	Blockspeicher	0	0	0	0
	Datenbank	0	2	0	4
	Kalender	2	8	16	16
	Kontaktdatenverwaltung	2	8	16	16
	Modularität	2	0	0	0
	Monitoringdienst	2	0	0	0
	Netzwerkdienst	0	2	0	4
	Objektspeicher	0	2	0	4
	Plattform	2	10	20	20
	Weboberfläche	2	0	0	0
Nicht-funktionale Kriterien	Aufwand	2	5	10	10
	Bedienung	2	4	8	8
	Benutzerhandbuch	2	0	0	0
	Durchsatz	2	1	2	2
	Kapazität	2	1	2	2
	Letztes Release	2	1	2	2
	Look and Feel	2	5	10	10
	Releasefrequenz	2	4	8	8
	Sicherheit	1	5	5	10
	Sprache der Dokumentation	1	0	0	0
	Vorhandensein von FAQ	2	2	4	4
	Wartungsanforderungen	2	6	12	12
	Zugriffsanforderungen	1	5	5	10
	Anzahl der Contributer	2	4	8	8
	Anzahl der Externen Provider	1	3	3	6

Abb. 19: Freund – Detailansicht ownCloud nach dem zweiten Gewichtungsschritt

Vergleich der Ergebnisse

Die vollständigen Ergebnisse der Auswertungen sind Anhang 7 und Anhang 8 zu entnehmen.

Der große Unterschied zwischen den Ergebnissen überrascht nicht. Als Auslöser können zwei zentrale Unterschiede identifiziert werden.

Zunächst stehen die gewünschten Anwendungsbereiche im Vordergrund. ownCloud unterscheidet sich hier stark von den anderen untersuchten Produkten, da es als Alleinstellungsmerkmal bereits eine fertige Kontaktdaten- und Kalenderverwaltung mitbringt. Diesen Vorteil bezahl der Nutzer jedoch mit einer geringeren Flexibilität. Im Gegensatz zu den Konkurrenzprodukten besitzt ownCloud keine so flexible und erweiterbare Infrastruktur, die nahezu jede beliebige Anwendung zulässt. So muss etwa auf die Funktionalität von Block- und Objektspeicher sowie Netzwerkdienst verzichtet werden. In den für ownCloud vorgesehenen Anwendungsgebieten stellt das jedoch keinen Nachteil dar.

Als wichtiges Entscheidungskriterium für ein Produkt stellt sich zudem die Unternehmensgröße heraus. Insbesondere für kleinere Unternehmen mit weniger IT-Ressourcen stellt die einfache Installation und Wartung ein wichtiges Kriterium dar. Während ownCloud bereits auf einem einfachen Linux-Server lauffähig ist, der auch von kleineren Betrieben bereitgestellt werden kann, benötigen die anderen untersuchten Produkte einen Hypervisor als Basis. Während die Bereitstellung einer derartigen Plattform für ein mittleres oder großes Unternehmen kein Problem ist, kann sie für ein kleines Unternehmen bereits eine große Herausforderung darstellen.

Damit wird deutlich, dass es im Private Cloud-Bereich nicht das Produkt gibt, welches in allen Bereichen stark ist und für jeden Einsatzzweck empfohlen werden kann. Allerdings ist auffällig, dass die gebotene Funktionalität von Eucalyptus, CloudStack und OpenStack sich ähneln. In dieser Gruppe sticht besonders OpenStack hervor, das mit seinen zahlreichen Modulen und der sehr großen Entwicklergemeinde punkten kann.

Erweiterungsmöglichkeiten des Tools

Die Erstellung des Tools zur Produktempfehlung stellt kein zentrales Ziel dieser Ausarbeitung dar. Es dient allein der einfacheren Auswertung des Kriterienkatalogs. In Rahmen einer Verbesserung, beispielsweise durch ein umfangreicheres Projekt, sind folgende Erweiterungen denkbar:

Workflow: Der Benutzer sollte gezielt durch das Programm geleitet werden. Dies beinhaltet eine Aufforderung der Eingabe zur Gewichtung inkl. Hilfebutton zur Beschreibung der einzelnen Kriterien und der empfohlenen Gewichtung. Erst nach Ausfüllen aller Kriterien sollte dann durch Klicken eines Buttons das Ergebnis angezeigt werden. Es gibt die Möglichkeit, die Erhöhung der funktionalen Kriterien bei Bedarf automatisiert ablaufen zu lassen. Nach der Anzeige der Ergebnisse bietet es sich an auf die Detailansichten durch einen Klick auf das jeweilige Tool zu verlinken. Ein Zurück-Button wäre an dieser Stelle für eine einfache Navigation von Nutzen.

Barrierefreiheit: Das Tool ist aktuell nicht barrierefrei und müsste für Menschen mit Behinderungen optimiert werden. Darüber hinaus sollten auch sonstige Gebote der Software-Ergonomie beachtet werden.

Layout: Die Ansicht in Excel ist in optischen Aspekten verbesserbar. Hier bietet es sich an, eine eigenständige Oberfläche zu programmieren, beispielsweis mittels Java.

Erweiterbarkeit: Eine Erweiterung um weitere Private Cloud-Produkte ist aktuell nur mittels eines gewissen Aufwands möglich. Die entsprechenden Formeln müssten manuell angelegt werden. An dieser Stelle bietet es sich an, eine Eingabemöglichkeit für Daten anderer Private Cloud-Produkte anzubieten, welche diese automatisiert einpflegt.

9 Exemplarische Realisierung einer Private Cloud am Beispiel von ownCloud

Autoren: Amelie Böer, Thomas Enderle, Matthias Holzwarth, Dominik Schrahe

Um auch praktische Erfahrungen wenigstens mit einem der vorgestellten Produkte sammeln zu können, soll in diesem Kapitel eine Realisierung vorgenommen werden. Aufgrund des begrenzten Umfangs der Arbeit musste hierfür ein Produkt ausgewählt werden. Nachdem OpenStack und ownCloud, wie bereits in Kapitel 5 Vorstellung ausgewählter Produkte dargestellt, zu den besonders herausragenden Open Source-Projekten des letzten Jahres zählen fiel die engere Wahl auf diese beiden Produkte. Da sich eine Realisierung von OpenStack sehr komplex gestaltet, wurde als Produkt zur exemplarischen Umsetzung die Community Edition von ownCloud ausgewählt. Im Folgenden soll die Installation sowie die sinnvolle Einrichtung für den Unternehmenseinsatz, mit besonderem Fokus auf die Sicherheit, auf einem Linux Debian-Server dargestellt werden. Anschließend erfolgt ein praktischer Test der Funktionen, um ownCloud auf die Alltagstauglichkeit in einem Versicherungsunternehmen zu untersuchen und etwaige Schwachstellen aufzudecken. Die für den professionellen Einsatz ebenfalls interessante und auf der Community Edition basierende Enterprise Edition konnte leider nicht getestet werden, da diese nicht unter Open Source-Lizenz zur Verfügung steht. In einem kurzen Abschnitt sollen dennoch die Unterschiede zur getesteten Community Edition aufgezeigt werden.

9.1 Installation und grundlegende Konfiguration

Da ownCloud eine bestehende LAMP-Konfiguration aus Linux-Betriebssystem, Apache-Webserver, MySQL-Datenbank und PHP voraussetzt, sind zunächst diese Komponenten auf dem Debian-System zu installieren. Alternativ für Apache und MySQL können auch Nginx oder SQLite als Webserver sowie Datenbank verwendet werden. Da der Apache Webserver in der _Versicherung bereits verwendet wird und SQLite vor allem für schwache Hardware und geringe Anforderungen im Betrieb optimiert ist, sollen Nginx und SQLite im Rahmen dieses Tests nicht weiter betrachtet werden.

Im nächsten Schritt ist die aktuelle Version 7.0.3 von ownCloud zu installieren, die, wie alle bereits genannten Komponenten, über die Debian-Repositories zur Verfügung steht. Bei der Einrichtung ist zunächst darauf zu achten, dass der Webserver im ownCloud-Verzeichnis, etwa /var/www/owncloud, Lese- und Schreibberechtigungen bekommt. Darüber hinaus muss zwangsweise die Datenbank vorkonfiguriert werden. Hierzu sollte in MySQL ein Benutzer sowie eine Datenbank für die ownCloud angelegt werden.

Es empfiehlt sich bereits während der Einrichtung das WebDAV-Modul von Apache zu aktivieren, das insbesondere für den Datenaustausch mobiler Endgeräte mit der ownCloud benötigt wird. Dies geschieht durch den Befehl

```
# a2enmod dav_fs
```

im Terminal. Anschließend kann bereits das Webinterface der ownCloud aufgerufen werden über das die Grundkonfiguration abgeschlossen wird. Dies geschieht, sofern das ownCloud-Verzeichnis an der oben genannten Stelle eingerichtet wurde, über die Adresse http://[ownCloud-Server-IP]/owncloud. Beim ersten Aufruf muss die Adresse des Datenverzeichnisses, etwa /var/www/owncloud/data/, sowie die verwendete Datenbank mit ihren Zugangsdaten angegeben werden. Zudem wird ein Administrator-Account angelegt.

Bevor eine tiefergehende Einrichtung per Weboberfläche vorgenommen wird, sollten einige grundsätzliche Konfigurationen vorgenommen werden. So reicht etwa für den sinnvollen Einsatz der ownCloud die standardmäßig von PHP vorgegebene maximale Dateigröße von 2 MB nicht aus. Folglich sollte in der unter /etc/php5/apache2 liegenden php.ini der Wert von "upload_max_filesize" beispielsweise auf "8000M" geändert werden. Gleichzeitig muss auch "post_max_size" auf "8000M" gesetzt werden. Damit lassen sich jedoch noch nicht bis zu rund acht GB große Dateien ablegen, da ownCloud standardmäßig nur Uploads mit 513MB Größe akzeptiert. Dieser Wert kann jedoch in der .htaccess-Datei im ownCloud Stammverzeichnis, in der Testinstallation also /var/www/owcloud, geändert werden. Hierfür sind die beiden Werte "php_value upload_max_filesize" und "php_value post_max_size" ebenfalls auf "8000M" zu setzen.

In einem letzten Schritt sollte noch die config.php im Verzeichnis /var/www/owncloud/config/ angepasst werden. In dieser sind die konfigurierten Datenbankzugangsdaten, samt Passwort im Klartext gespeichert. Zudem ist das Verzeichnis zur Ablage von Daten hinterlegt. Bei zukünftigen Änderungen dieser Einstellungen sind hier folglich Anpassungen vorzunehmen. Für die initiale Einrichtung ist vor allem das „trusted_domains"-Array anzupassen. In diesem Array sind alle IP-Adressen oder Hostnamen zu definieren, unter denen die ownCloud-Instanz erreichbar sein soll. Wird diese von einer nicht in besagtem Array definierten Adresse aus aufgerufen, erscheint eine Fehlermeldung und die Anmeldung ist nicht möglich.

Weitergehende Konfigurationen können anschließend über die Weboberfläche vorgenommen werden.

9.2 Konfigurationsoptionen über das Webinterface

Für Administratoren steht in der Weboberfläche unter "Benutzer" eine Benutzerverwaltung zur Verfügung. Unter "Administration" können sie diverse Einstellung verändern. Neben diesen, nach einem Klick auf den Benutzernamen sichtbaren Schaltflächen, stellt die App-

verwaltung wichtige Funktionen bereit. Abb. 20 zeigt das Webinterface von ownCloud und wie die genannten Funktionen zu erreichen sind. Zu beachten ist, dass die rot umrahmten Einstellungsmöglichkeiten nur für Administratoren sichtbar sind.

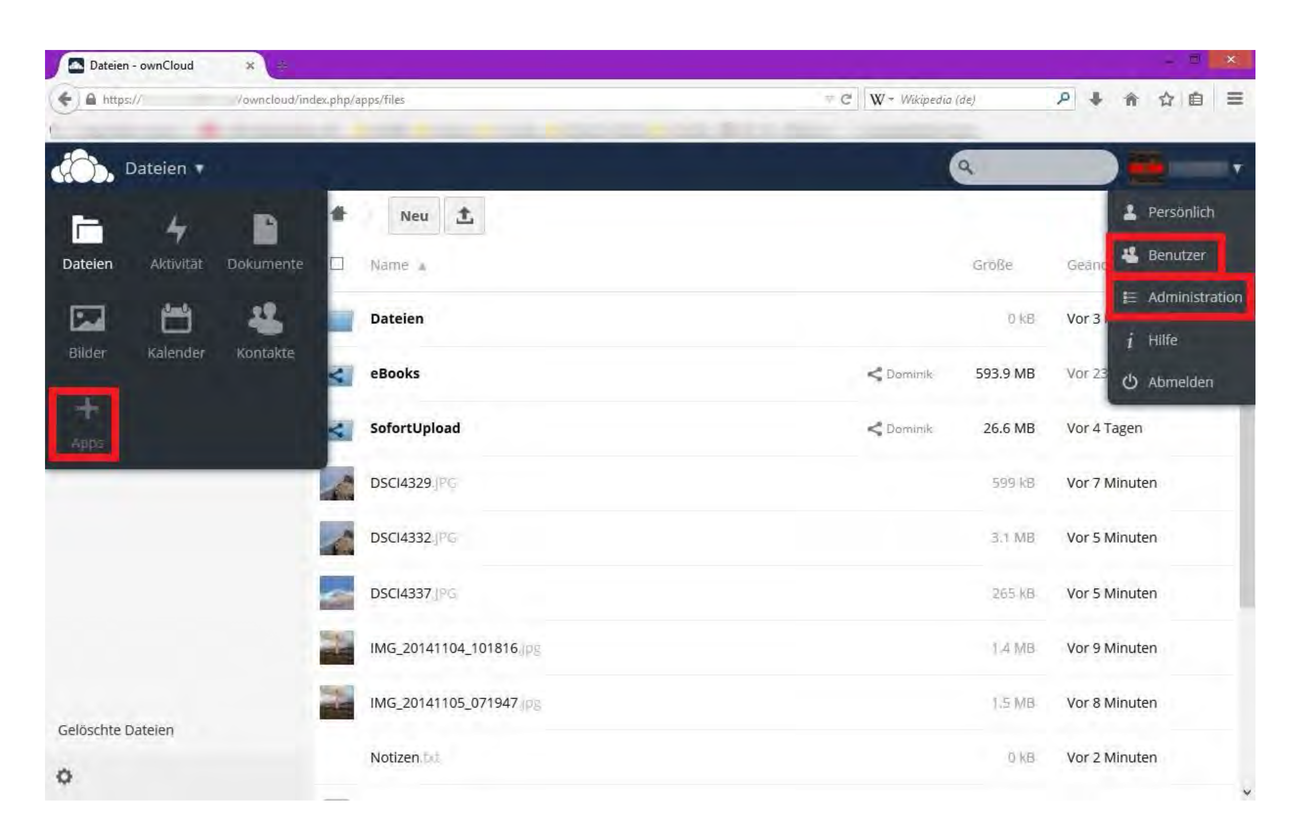


Abb. 20: Das Webinterface von ownCloud. Markiert sind die nur für Administratoren sichtbaren Funktionen zur Konfiguration der ownCloud

Viele Funktionalitäten der ownCloud sind als Apps realisiert. Einige dieser Apps sind bereits installiert und müssen lediglich noch aktiviert werden. Andere Apps wie Calendar oder Contacts zur Kalender- und Kontakteverwaltung sind bereits aktiv. Wie einfach das aktivieren und deaktivieren von Apps möglich ist, wird auch auf dem Screenshot der App-Verwaltung deutlich, der sich in Anhang 9 befindet.

Sollen die Benutzer Dokumente in den Open Document oder Microsoft Office-Formaten gemeinsam über den Browser bearbeiten können, muss die „Documents"-App aktiviert werden. Diese enthält standardmäßig eine einfache Bearbeitungsfunktion für Open Document Text Dateien. Sollen auch andere Formate angezeigt und bearbeitet werden können, so muss auf dem ownCloud Server LibreOffice installiert werden.

Sofern weitere Speicher angebunden werden sollen, kann die App „External Storage Support" aktiviert werden. Hierüber lassen sich nicht nur Public Cloud-Speicherdienste wie Dropbox, Google Drive oder Amazon Simple Storage Service einbinden, sondern auch FTP-Server. An dieser Stelle sei noch erwähnt, dass neben dem sekundären Datenspeicher auch für die primäre Datenspeicherung, die Standardmäßig auf das ownCloud Unterverzeichnis /data zurückgreift, bestehende Systeme verwendet werden können. Unterstützt werden unter

anderem GlusterFS oder OpenStack Swift. Letzteres bietet sich an, wenn die Funktionalität von ownCloud benötigt wird, und eine OpenStack-Installation bereits im Unternehmen besteht.

Neben den zahlreichen bereits nach der Installation vorinstallierten Apps besteht unter apps.owncloud.com auch ein Appstore, über den weitere Anwendungen bereitgestellt werden.

Nach der Aktivierung von Apps erscheinen in der Regel im Administrationsbereich neue Einstellungsmöglichkeiten. So kann für die Bearbeitungsfunktion von Office-Dokumenten konfiguriert werden, ob die Darstellung mittels einer lokal installierten Open- bzw. Libre-Office-Instanz oder über einen externen Server erfolgen soll.

Darüber hinaus kann die Funktion zum Teilen von Dateien im Administrationsbereich konfiguriert werden, um beispielsweise nur das Teilen mit anderen ownCloud-Benutzern zu erlauben. Alternativ können Datei- und Ordnerfreigaben per Link an Dritte verpflichtend mit einem bei der Freigabe zu definierenden Passwort geschützt werden. Daneben empfiehlt sich die Einrichtung der Mail-Server Funktion. Über diese können einerseits geteilte Links automatisiert verschickt werden. Andererseits lassen sich so vergessene Benutzerpasswörter ohne zusätzlichen Aufwand eines Administrators durch den Benutzer zurücksetzen. Sofern ein Unternehmen mehrere ownCloud-Installationen, beispielsweise für verschiedene Abteilungen, nutzt oder Daten mit Partnerunternehmen austauschen will, kann das Server-to-Server-Sharing aktiviert werden. Hierüber können mehrere ownCloud-Installationen miteinander verbunden werden.

Nach dieser Grundlegenden Einrichtung per Webinterface kann der Administrator im Anschluss erste Benutzeraccounts für die produktive Nutzung anlegen. Benutzer können hierbei in Gruppen eingeteilt werden, die beispielsweise Abteilungen oder Projektgruppen im Unternehmen entsprechen können.

Anschließend können Anwender Verzeichnisse nicht mehr nur mit einzelnen Benutzern sondern auch mit Gruppen teilen. Weitere, durch die Definition von Gruppen zu erreichende, Mehrwerte werden im Kapitel 9.4 Funktionen von ownCloud im produktiven Einsatz im Unterabschnitt Webinterface deutlich.

Daneben sind Begrenzungen der Speichervolumina in der Benutzerverwaltung möglich. Sofern die Benutzerverwaltung nicht manuell über die ownCloud erfolgen soll, kann auch ein

Active Directory[257] oder LDAP-Backend benutzt werden, über das eine bestehende Benutzer- und Gruppenstruktur im Unternehmen für die ownCloud verwendet werden kann.

9.3 Einrichtung vorhandener Sicherheitsfunktionen und Absicherung gegen aktuelle Bedrohungen

Da bereits einleitend auf die besondere Relevanz der Sicherheit bei Private Clouds, insbesondere beim Einsatz in Versicherungen, eingegangen wurde, soll in diesem Abschnitt die Absicherung einer ownCloud-Installation im Speziellen betrachtet werden. Die im Folgenden erläuterten Einstellungen besitzen besondere Relevanz, sofern die ownCloud auch aus dem Internet erreichbar sein soll, dennoch erscheint die Anwendung vieler Optionen auch bei einer Nutzung ausschließlich im lokalen Netzwerk sinnvoll.

Zunächst bietet es sich bereits währen der Konfiguration des Webservers an, ein Zertifikat für die Domain, unter der der Server verfügbar sein soll, zu hinterlegen. Dieses stellt die Voraussetzung für die Verwendung einer Trasportverschlüsselung und damit der sichere Datenübertragung zwischen Server und Client im Betrieb dar. Hierfür muss zunächst durch den Befehl

```
# a2enmod ssl
```

das SSL-Modul des Apache-Webservers aktiviert werden. Anschließend können das Zertifikat und der Schlüssel, beispielsweise owncloud.crt und owncloud.key etwa in einem Unterverzeichnis /SSL im Apache2-Verzeichnis abgelegt werden. Damit der Webserver diese verwendet, muss im Unterverzeichnis sites-available des Apache Webservers die default-ssl-Datei angepasst werden. Hierfür müssen in den Zeilen „SSLCertificateFile“ und „SSLCertificateKeyFile“ die Pfade zu den entsprechenden Dateien nach dem Schema

```
SSLCertificateFile /etc/apache2/SSL/owncloud.crt

SSLCerfificateKeyFile /etc/apache2/SSL/owncloud.key
```

eingetragen werden.

In der Regel ist es sinnvoll, lediglich eine verschlüsselte Kommunikation anzubieten. Hierzu kann im Administrationsbereich auf der Weboberfläche der ownCloud unter Sicherheit „HTTPS erzwingen“ aktiviert werden. Dadurch wickelt der Server nicht nur Browserverbindungen ausschließlich verschlüsselt ab, sondern erlaubt auch Desktop Clients, Smartphone Apps und per WebDAV zugreifenden Anwendungen, ausschließlich verschlüsselte Verbin-

[257] Active Directory ist der Name des Verzeichnisdienstes, der von Mictrosoft Windows Server-Betriebssystemen zur Verfügung gestellt wird. Hierüber lassen sich unter anderem Zugriffsrechte für Benutzer und Gruppen verwalten.

dungen. Dabei kann die Art der Verschlüsselung weiter konfiguriert werden, indem entsprechende Einstellungen im Apache Webserver vorgenommen werden. In der Datei ssl.conf, die sich im Verzeichnis /etc/apache2/mods-enabled befindet, kann beispielsweise bestimmt werden, dass sämtliche Verbindungen AES256-verschlüsselt abgewickelt werden. Hierfür muss lediglich die Zeile

```
SSLCipherSuite HIGH:MEDIUM: !aNULL:!MD5
```

in

```
SSLCipherSuite AES256+EECDH
```

geändert werden. Dabei erhöhen sich zwar die Ressourcenanforderungen im Bezug auf Rechenleistung sowohl Server- als auch Clientseitig, gleichzeitig wird aber auch die Verbindungssicherheit auf aus heutiger Sicht optimales Niveau gehoben.

Nach Vornahme dieser Einstellung baut der ownCloud-Server nur noch AES-256 verschlüsselte TLS-Verbindungen auf. TLS ist ein Protokoll, das verschlüsselte Verbindungen ermöglicht.[258] Für den Schlüsselaustausch wird Elliptic Curve Diffie-Hellman Ephemeral verwendet. Zur Verschlüsselung des AES-Keys wird RSA eingesetzt. Durch die Kombination der verschiedenen Verfahren ergibt sich eine Verschlüsselung, die gegen die meisten heute bekannten Angriffe gesichert ist.[259] Weitere Hintergründe werden im Abschnitt 6.2 Verschlüsselung erläutert. Onlinebanking-Portale wie die der Volks- und Raiffeisenbanken sowie die der Sparkassen verwenden dasselbe Verschlüsselungsverfahren.

Neben dieser Transportverschlüsselung bietet ownCloud eine Dateiverschlüsselung auf dem Server an. Diese wird als eigenständige, “Encryption” genannte App bereitgestellt, die jedoch bereits installiert ist und lediglich unter Apps vom Administrator aktiviert werden muss.

Nach der Aktivierung werden beim nächsten Zugriff eines Benutzers auf sein Konto sowie beim Upload neuer Dateien alle Daten verschlüsselt abgelegt. Auf diese Weise ist es selbst Personen mit root-Zugriffsrechten auf den Server nicht möglich, gespeicherte Daten auszulesen. Damit sollten die Daten auch gegen Hackerangriffe ausreichend gesichert sein. Diese serverseitige Datenverschlüsselung gewinnt insbesondere mit der Benutzung externer Speicher, die im Kapitel 9.2 Konfigurationsoptionen über das Webinterface bereits angedeutet wurde, an Bedeutung. In diesem Fall werden hochgeladene Dateien zunächst verschlüsselt und dann beispielsweise in einer Public Cloud abgelegt. Bei der Dateiverschlüsselung handelt es sich um eine 128 Bit AES Verschlüsselung, wobei das Benutzerpasswort als Schlüs-

[258] Vgl. Network Working Group (2008)
[259] Vgl. Bundesamt für Sicherheit in der Informationstechnik (2013)

sel verwendet wird.[260] Daher ist bei Benutzung dieser Funktion die Sensibilisierung der Benutzer für die Verwendung sicherer Passwörter besonders wichtig. Der Nachteil dabei besteht darin, dass der Zugriff auf die Daten unmöglich wird, sofern der Benutzer sein Passwort vergisst. Um einen solchen Datenverlust zu vermeiden, kann durch den Administrator ein Wiederherstellungsschlüssel definiert werden, über den Daten alternativ wiederhergestellt werden können. Hierbei ist zu beachten, dass jeder Benutzer individuell in seinen persönlichen Einstellungen die Verwendung dieses Schlüssels aktivieren muss.

Sowohl die Transport- als auch die serverseitige Dateiverschlüsselung können jedoch keine vollständige Sicherheit bieten. Die Ablage besonders sensibler Dateien sollte also weiterhin nicht in der Cloud erfolgen, zumal auch das Risiko von gutgläubigen Benutzern besteht, wie erst jüngst durch eine Studie bewiesen wurde.[261] Zudem ist, wie bei jedem Webserver, eine permanente aktive Wartung erforderlich. So muss durch die Installation aktueller Updates sichergestellt werden, dass keine veralteten und auf Sicherheitslücken anfällige Programmversionen verwendet werden. Auf diese Weise kann der Server gegen Sicherheitslücken wie Heartbleed oder ShellShock abgesichert werden. Daneben empfiehlt sich zum aktuellen Zeitpunkt eine manuelle Absicherung gegen Poodle-Angriffe, bei denen Angreifer eine SSLv3-Verschlüsselung erzwingen, die heute nur noch geringen Schutz bieten kann. Da alle modernen Browser und Clients auch das deutlich sichere TLS, wenigstens in der Version 1.0 unterstützen, ist die Verwendung von SSLv3 nicht mehr zeitgemäß. Einige Browser wie Mozilla Firefox blockieren ab Version 34 bereits Clientseitig derart verschlüsselte Verbindungen. Auch Google plant mit der Version 30 von Chrome die Deaktivierung von SSLv3.[262] Um jedoch auch die Verbindungen mit anderen Browsern beziehungsweise älteren Versionen abzusichern, empfiehlt sich eine Serverseitige Blockierung von SSLv3 Verbindungen. Hierzu muss die im Verzeichnis /etc/apache2/mods-enabled gespeicherte Datei ssl.conf bearbeitet werden. Die Zeile

```
SSLProtocol all -SSLv2
```

deutet bereits an, dass der Server alle Protokolle unterstützt und nur Verbindungen mit dem veralteten SSLv2 zurückweist. Darüber werden durch Ergänzen von “-SSLv3” zu

```
SSLProtocol all -SSLv2 -SSLv3
```

nach einem Neustart des Apache Webservers per

```
# sudo service apache2 restart
```

[260] Für detaillierte Informationen vgl.: Schießle, B. (2013)
[261] Vgl. dazu ausführlich: Sokolov, D. A. J. (2014)
[262] Vgl. Eikenberg, R. (2014)

keine SSLv3-Verbindungen mehr aufgebaut. Damit sind Verbindungen auch zu älteren Browserversionen gegen Poodle-Angriffe abgesichert.

Die automatische Dateisynchronisation stellt an sich bereits ein gewisses Sicherheitsrisiko dar. Insbesondere wenn externe Clients angebunden werden, die sich nicht im besonders gesicherten Unternehmensnetzwerk befinden, könnten Schadprogramme automatisiert auf diverse Geräte synchronisiert werden. Um dies zu vermeiden ist in ownCloud bereits der Virenscanner von ClamAV integriert, der sämtliche hochgeladene Dateien überprüft.[263]

Die nachfolgenden Tests mit der ownCloud-Installation wurden mit den hier und weiter oben beschriebenen Einstellungen vorgenommen.

9.4 Funktionen von ownCloud im produktiven Einsatz

Webinterface

Das Webinterface der ownCloud ist unter https://[Domain]/[owncloud-Pfad auf dem Server] erreichbar, in der für diese Arbeit benutzten Testinstallation also https://[Domain]/owncloud. Hierüber können sämtliche Funktionen, die die ownCloud zur Verfügung stellt, genutzt werden. Die im weiteren Verlauf dieses Kapitels beschriebenen Synchronisations-Clients für diverse Geräte sind also prinzipiell nicht notwendig, können jedoch den Komfort bei der Nutzung erheblich erhöhen. Aufgrund des begrenzten Umfangs der vorliegenden Ausarbeitung, soll im Folgenden nur auf die nicht selbsterklärenden sowie besonders interessanten Funktionen des Webinterface eingegangen werden.

Dateien und Ordner können jeweils heruntergeladen werden, wobei im Falle von Ordnern ein .zip-Archiv des Ordners erstellt wird. Darüber hinaus ist auch das Teilen, also zur Verfügung stellen von Dateien oder Verzeichnissen für andere Benutzer oder Dritte möglich. Hierfür können Benutzer- oder Gruppennamen eingetragen werden. Alternativ ist das Teilen über einen Link möglich. Hierbei kann ein Passwort vergeben und ein Ablaufdatum des Links gesetzt werden. Die beim Teilen einer Datei möglichen Optionen werden auch aus dem Screenshot in Abb. 21 ersichtlich.

[263] Vgl. ownCloud (2014b)

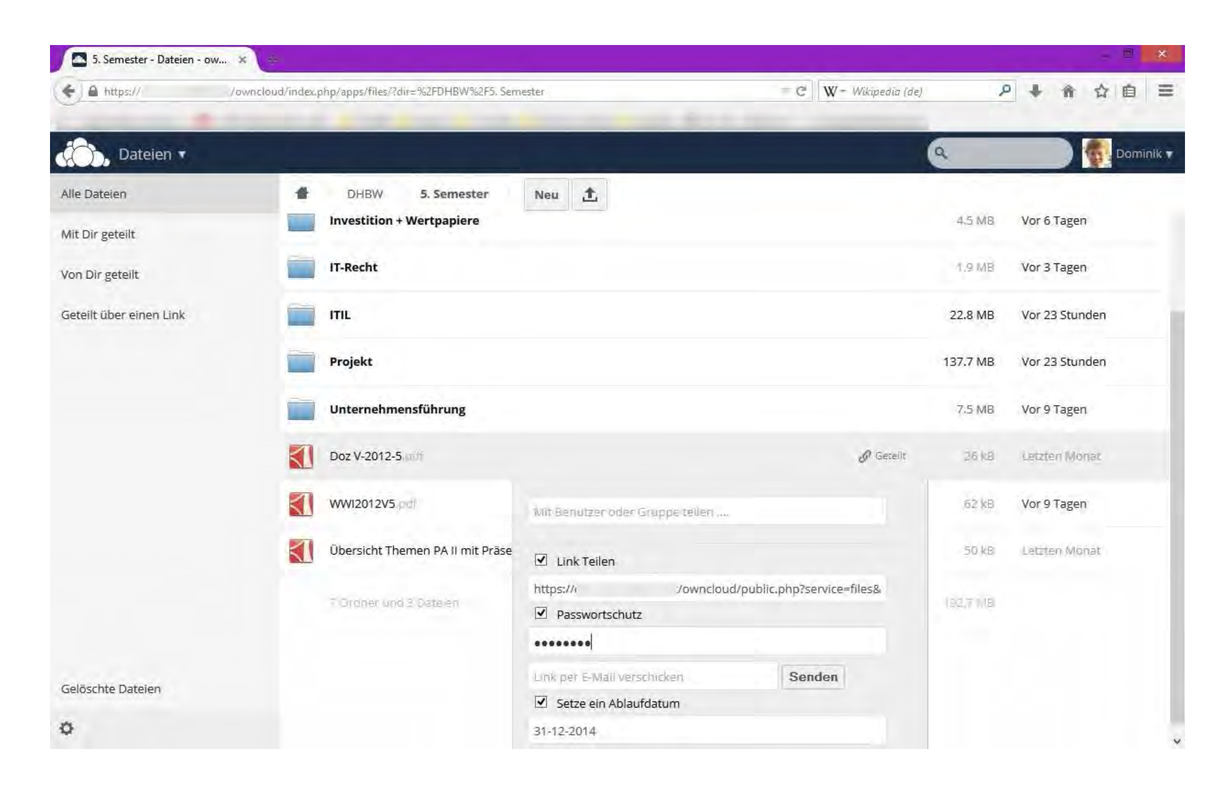


Abb. 21: Optionen beim Teilen einer Datei: Teilen mit Benutzern oder per Link, dann wahlweise mit Passwortschutz und Ablaufdatum. Außerdem kann der Link per Mail an einen beliebigen Empfänger gesandt werden

Beim Teilen eines Ordners kann darüber hinaus Dritten das Hochladen von Dateien in den freigegebenen Ordner sowie die Bearbeitung freigegebener Dateien erlaubt werden. Das hierbei Daten verloren gehen wird durch einen Dateiversionsverlauf ausgeschlossen. So speichert ownCloud standardmäßig alte Versionen von Dokumenten, sodass diese zu einem späteren Zeitpunkt angesehen oder wiederhergestellt werden können. Bilder und PDF-Dokumente können in einem internen Viewer angezeigt werden, während sich .txt-Dokumente direkt bearbeiten lassen. Dasselbe gilt, wie bereits im Kapitel 9.2 Konfigurationsoptionen über das Webinterface beschrieben, für Office-Dokumente, sofern die entsprechende App installiert ist. In diesem Fall kann auch eine gleichzeitige Bearbeitung von Dokumenten erfolgen, wobei jeder Bearbeiter unmittelbar sehen kann, welche Änderungen andere Bearbeiter vornehmen. Wie in Abb. 22 sichtbar, werden hierbei sämtliche Bearbeiter am Rand des Dokuments in einer charakteristischen Farbe dargestellt. In der gleichen Farbe wird der Cursor des jeweiligen Benutzers angezeigt, wobei Änderungen unmittelbar sichtbar werden.

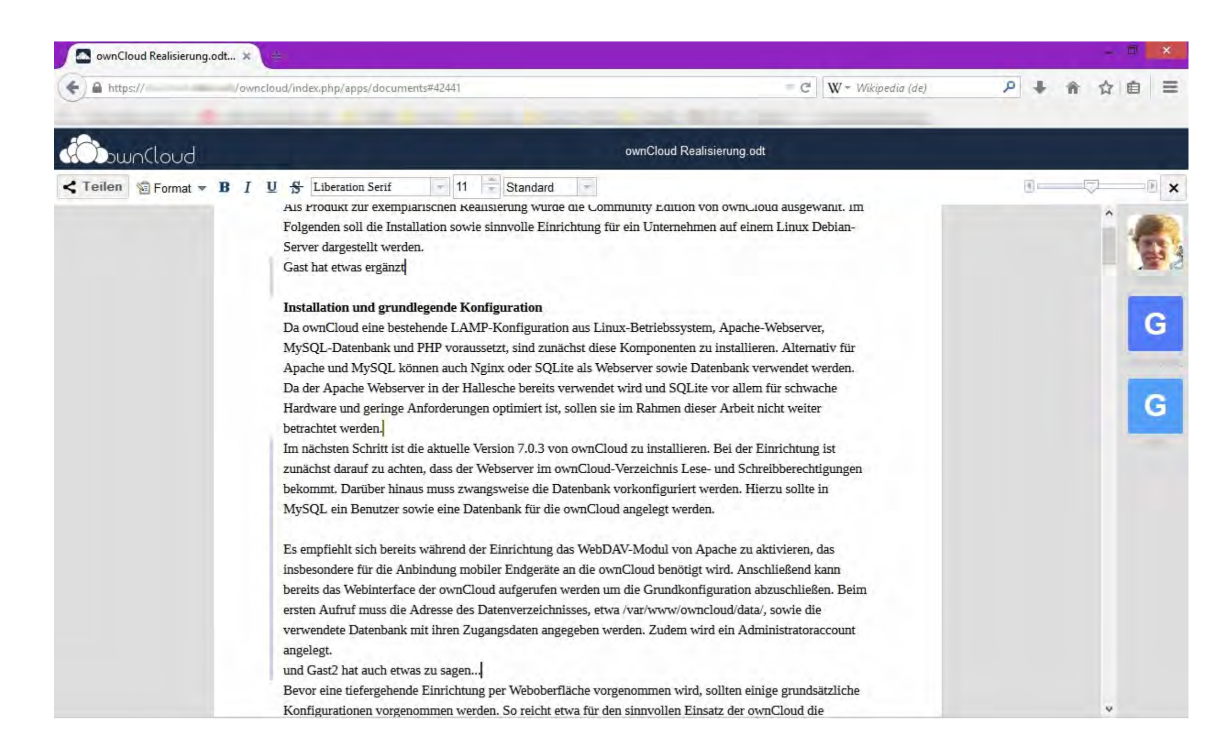


Abb. 22: Bearbeitung eines .odt-Dokuments im Browser mit mehreren Personen

Neben der Dateiverwaltung und -bearbeitung lassen sich über das Webinterface auch Kontakte verwalten. Kontakte können in Gruppen organisiert werden, auch der Export in das .vcf-Dateiformat ist möglich. Umgekehrt lassen sich Kontakte aus dem vCard- und VCF-Format importieren. Zur Unterstützung der Zusammenarbeit zum Beispiel von Projektgruppen können Kontakte genau wie Dateien mit anderen Benutzern oder Gruppen geteilt werden. Ein Screenshot der Kontakteverwaltung im Webinterface zeigt Abb. 23.

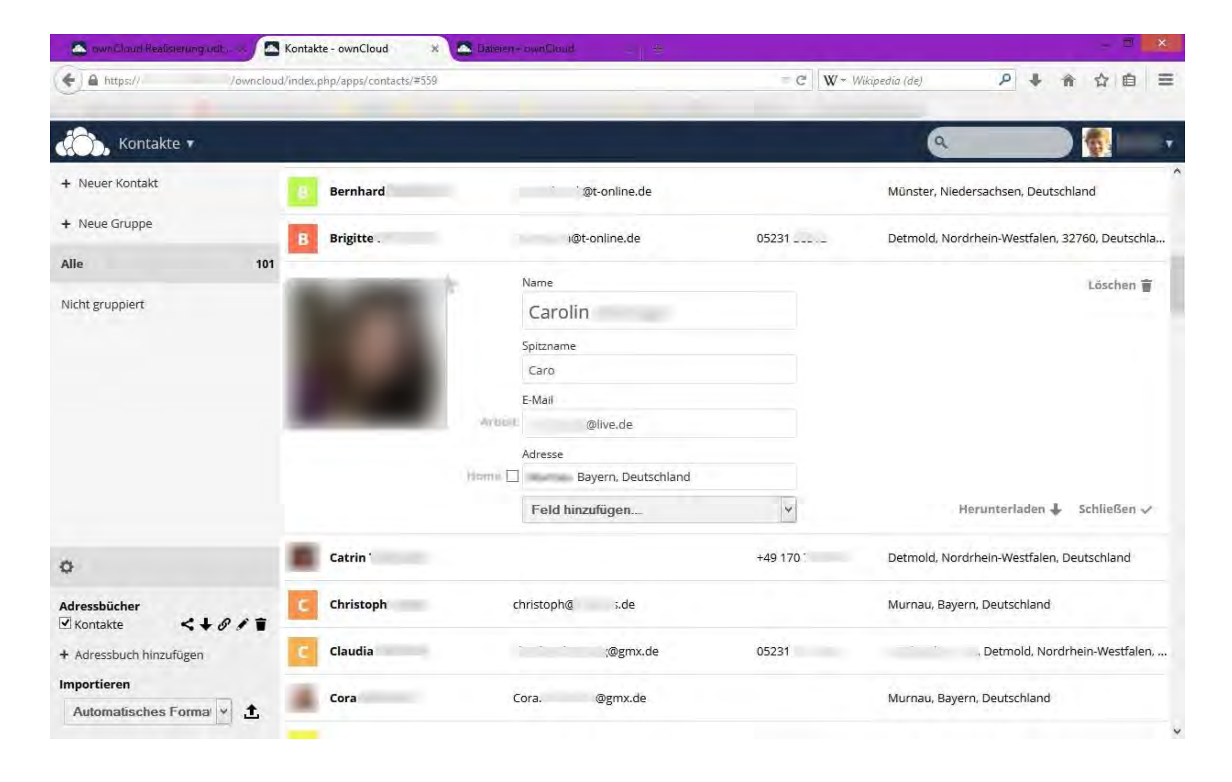


Abb. 23: Kontakte-Verwaltung im Browser

Darüber hinaus ist auch die Verwaltung von Kalendern und Terminen mit ownCloud möglich. Neben den üblichen Terminverwaltungswerkzeugen existiert hier ebenfalls die Möglichkeit, Termine für andere Personen oder Gruppen freizugeben. Der Screenshot in Abb. 24 zeigt die Bearbeitung eines Kalendereintrags.

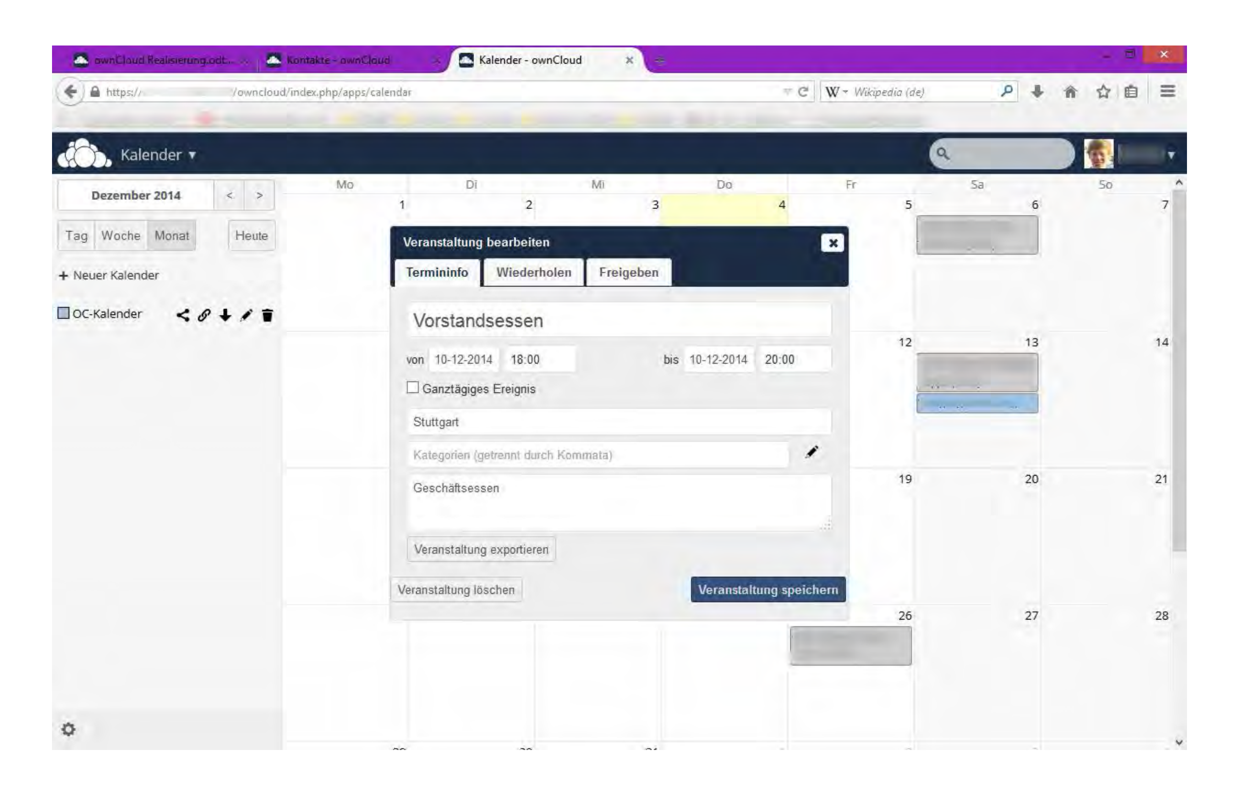


Abb. 24: Dialog zur Erstellung eines Termins, weitere Optionen finden sich unter den Reitern Wiederholen und Freigeben

Synchronisation mit Desktop-PCs

Neben den Umfangreichen Funktionen der Weboberfläche stellt insbesondere die Anbindung diverser Clients mit automatisierter Synchronisation ein zentrales Feature von ownCloud dar. Um Daten aus freigegebenen Verzeichnissen von Desktop-PCs synchronisieren zu können, werden Clients für Windows, Linux und Mac OS bereitgestellt. Diese können wahlweise alle oder nur bestimmte der auf dem Konto gespeicherten Verzeichnisse und Dateitypen mit der ownCloud synchron halten. Darüber hinaus bieten sie ähnliche Einstellungsmöglichkeiten wie gängige Clients von Public Cloud-Anbietern, zum Beispiel für Proxyserver und die Bandbreitenbegrenzungen. Ein Screenshot vom Client unter Windows befindet sich in Abb. 25.

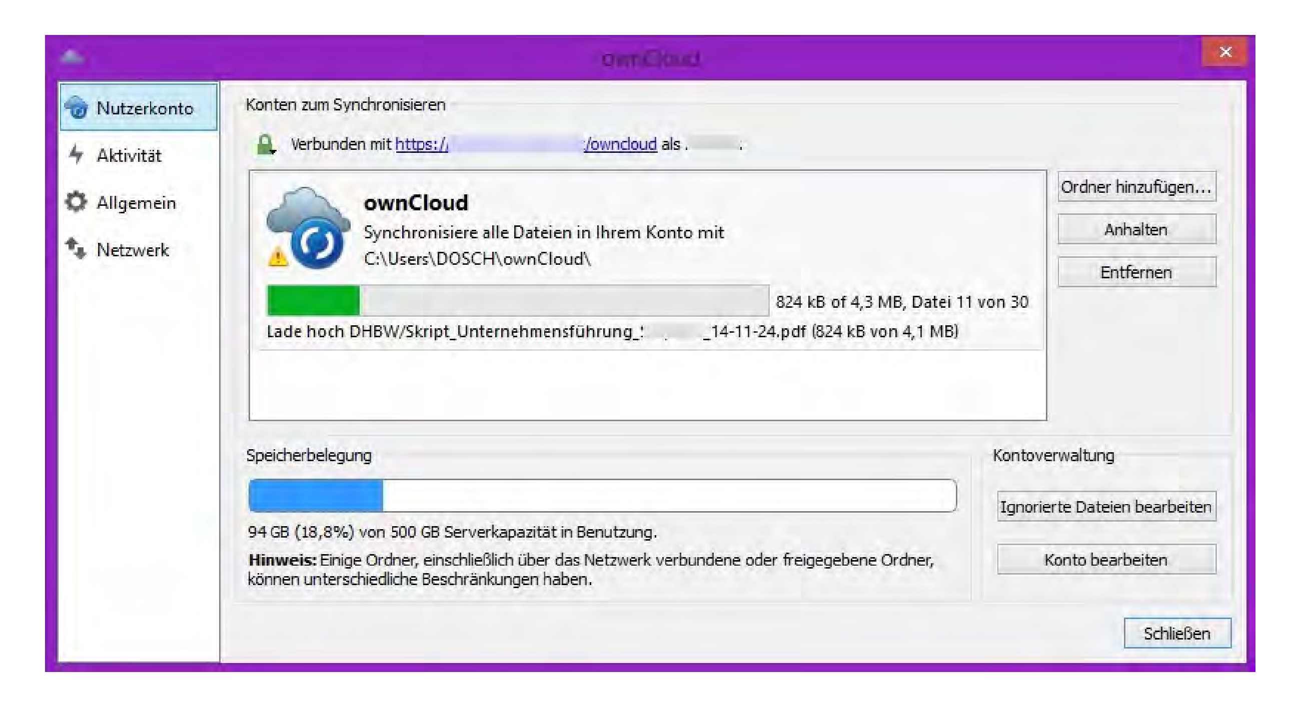


Abb. 25: ownCloud-Client unter Windows

Alternativ ist die Dateisynchronisation via WebDAV-Protokoll möglich, das von vielen Dateimanagern bereits unterstützt wird. Die WebDAV-Adresse, die in der Dateiverwaltung im Webinterface sichtbar ist, ist nach dem Schema https://[Domain]/owncloud/remote.php/webdav aufgebaut. Zu beachten ist, dass sich die Funktionalität der offiziellen Clients auf die Datei- und Verzeichnissynchronisation beschränkt. Zum Abgleichen von Kontakten und Kalendern sind weitere Maßnahmen notwendig, die im Folgenden beschrieben werden sollen.

Zu diesem Zweck werden CardDAV und CalDAV als Protokolle angeboten. Die Tests im Rahmen dieser Ausarbeitungen wurden mit Mozilla Thunderbird durchgeführt, da es sich hierbei um eine plattformübergreifende Lösung handelt, die ebenfalls unter Open Source-Lizenz steht. Die Einrichtung sollte folglich auf allen gängigen Betriebssystemen ähnlich ablaufen. Zunächst müssen in Thunderbird zwei Add-Ons installiert werden. Für den Abgleich von Kontakten wird der „SOGo-Connector“ benötigt. Nach der Installation kann im Adressbuch unter Datei > Neu > Remote-Adressbuch ein ownCloud Adressbuch importiert werden. Dabei muss die CardDAV-Adresse angegeben werden, die auf dem Webinterface im Kontakte-Bereich angezeigt wird und nach dem Schema https://[Domain]/owncloud/remote.php/carddav/adressbooks/[Benutzername]/[Adressbuchname]

aufgebaut ist. Zudem kann die Synchronisationshäufigkeit bestimmt und weitere Einstellungen vorgenommen werden. Nach einem Neustart Thunderbirds erfolgt die erste Synchronisation, wofür SOGo die Zugangsdaten erfragt. Für die Kalendersynchronisation wird das Add-On „Lightning“ benötigt. Anschließend kann über „Neuer Kalender“ ein Kalender im

Netzwerk eingerichtet werden, der per CalDAV angebunden wird. Der CalDAV-Link ist nach dem Schema https://[domain]/owncloud/remote.php/caldav/calendars/[Benutzername]/[Kalendername] aufgebaut. Die Synchronisationshäufigkeit ist unter Eigenschaften für jeden auf die beschriebene Weise importierten Kalender einstellbar. Die Einstellungsmöglichkeiten für die Kontakte- und Kalendersynchronisation in Thunderbird unter Fedora sind in Abb. 26 und Abb. 27 abgebildet. Die Tests wurden ebenso unter Microsoft Windows 8.1 durchgeführt, wobei sich die Installation und Einrichtung nicht vom beschriebenen Vorgehen unterscheidet.

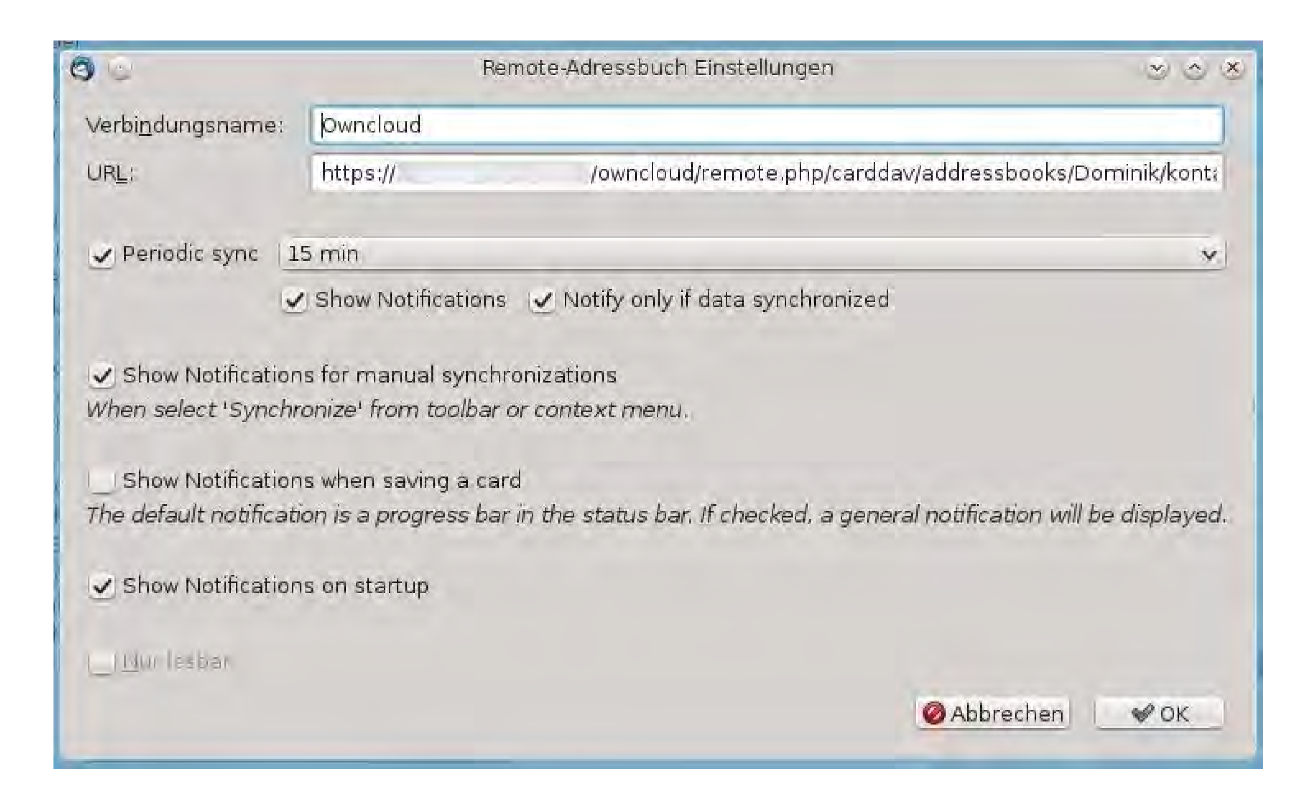


Abb. 26: Einstellungsmöglichkeiten des Mozilla Thunderbird-Plugins SOGo-Connector unter Linux

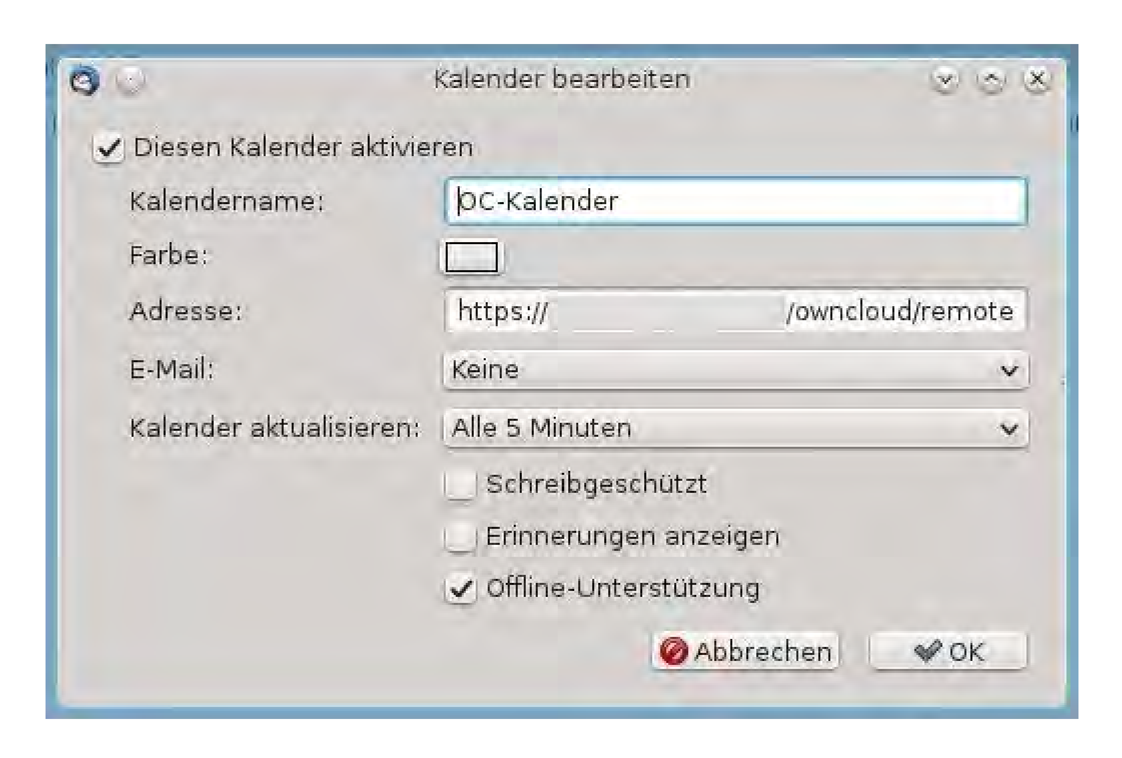


Abb. 27: Optionen des Mozilla Thunderbird-Plugins Lightning zur Kalendersynchronisation unter Linux

Synchronisation mit Smartphones und Tablets

Nachdem heute mobile Geräte im Unternehmen eine immer wichtigere Position einnehmen, stellt deren Anbindung auch bei der Einführung einer Private Cloud eine zentrale Anforderung dar. Zunächst existieren von der ownCloud Community entwickelte Apps für Android und iOS, mittels derer ein Zugriff auf gespeicherte Dateien möglich ist. Daneben bieten diese Funktionen um Dateien und Verzeichnisse, auch mittels QR-Code, zu teilen und Bilder sowie Videos anzuzeigen. Auch die automatisierte Synchronisation von Verzeichnissen, wie sie auf Desktop-Rechnern üblich ist, ist über die Apps durchführbar. Da die Einstellungen hierfür jedoch sehr beschränkt sind, wurde in dieser Ausarbeitung eine Synchronisationsapp mit WebDAV-Unterstützung gewählt. Die Tests erfolgten mittels FolderSync unter Android. Diese App bietet diverse Einstellungsmöglichkeiten. Für den professionellen Einsatz sollte vor allem festgelegt werden, ob mobile Endgeräte lediglich zur Anzeige oder auch zur Produktion von Inhalten verwendet werden sollen. Im ersten Fall ist anzuraten nur eine Einwege-Synchronisation zum Gerät zu werden. Darüber hinaus bietet FolderSync Einstellungen zum Synchronisationsintervall und den Netzwerken, über die die Synchronisation stattfinden soll. Auf diese Weise kann nicht nur mobiles Datenvolumen eingespart werden, sondern auch bestimmte WLAN-Nerzwerke definiert werden. So wäre es möglich, dass eine Synchronisation ausschließlich über das firmeninterne WLAN erfolgt. Darüber hinaus ist FolderSync mittels eines PIN-Codes gegen Veränderungen der Einstellungen durch Unbefugte absicherbar. Ein Ausschnitt der unter vornehmbaren Synchronisations-Einstellungen befindet sich in Anhang 10.

Bei der Bereitstellung von Kontakten und Kalendern muss zwischen iOS und Android-Geräten unterschieden werden. Bei der Verwendung von iOS-Geräten ist in den Einstellungen ein CardDAV- sowie CalDAV-Account hinzuzufügen. Anschließend werden Kontakte und Kalender automatisch in beide Richtungen auf dem aktuellsten Stand gehalten. Zu Beachten ist jedoch, dass hier eine andere CalDAV-Adresse als üblich, nach dem Schema https://[Domain]/owncloud/remote.php/caldav/principals/[Benutzername], verwendet werden muss. Diese ist jedoch im Webinterface einsehbar.

Android-Geräte benötigen zur Kontakte- und Kalendersynchronisation Zusatzapps, die jedoch kostenlos im Play-Store verfügbar sind. Die Kontaktesynchronisation kann über die App „CardDAV“ abgewickelt werden, für den Abgleich von Kalendern kann „CalDav SyncAdapter“ verwendet werden. Zur Einrichtung sind jeweils die Standard-CardDAV- sowie CalDAV-Adresse und die Zugangsdaten anzugeben. CardDAV Sync Adapter bietet zudem Möglichkeiten zur Konfiguration vom Synchronisationsintervall sowie den zu verwendenden Datendiensten.

9.5 Praktische Erfahrungen im Rahmen von Tests

Um neben den theoretischen Erläuterungen zur Einrichtung und den Funktionalitäten von ownCloud sowie kompatibler Synchronisations-Clients auch den tatsächlichen Nutzwert für Unternehmen bewerten zu können, wurden im Anschluss an die Installation und Konfiguration einige Tests durchgeführt. Hierfür wurden Testdaten, Kontakte und Kalender zwischen diversen Geräten synchronisiert. Bei ersterem wurde sowohl Wert darauf gelegt, dass sehr tiefe Verzeichnisstrukturen mit vielen kleinen Dateien als auch einige große Dateien synchronisiert wurden.

Grundsätzlich funktionierte die Synchronisation von Dateien zwischen Desktop-PCs, die die offiziellen ownCloud-Clients verwendeten sehr zuverlässig. Lediglich der Client unter Linux verlor nach dem Aufwachen aus dem Energiesparmodus in seltenen Fällen die Verbindung zum Server und musste anschließend manuell neugestartet werden, um die Synchronisationsaktivität wieder aufzunehmen. Die Reaktionszeiten lagen zwar im Vergleich zu bekannten Public-Cloud-Diensten wie Dropbox minimal höher, blieben aber in einem für den Arbeitsalltag angenehmen Bereich. So dauerte es nach dem Hinzufügen von Dateien auf einem Client einige Sekunden, bis dieser mit der Synchronisation begann und wiederum einige, bis die Datei auf einen anderen Client heruntergeladen wurde. Auch die Synchronisation von großen Dateien über einem Gigabyte verlief zuverlässig. Dabei fiel auf, dass ownCloud im Gegensatz zum Beispiel zu Dropbox keine blockbasierte Synchronisation verwendet. Während Dropbox nur die geänderten Teile einer Datei übertragen muss, synchronisiert ownCloud die gesamte Datei nochmal, was bei häufiger Speicherung zu deutlich höherem Traffic führt. Dies kann aber auch zu weiteren Problem führen, da auf Dateien, während sie durch den

ownCloud Desktop-Client synchronisiert weden, kein schreibender Zugriff möglich ist. Beim häufigen Speichern großer Dateien können somit Fehlermeldungen wie in Abb. 28 auftreten, was nicht benutzerfreundlich ist. Bei schmalbandigem Upload führt selbst die Speicherung eines mehrere MB großen Word-Dokuments im Minutentakt, was viele Benutzer bereits aus Gewohnheit machen, zu Zugriffsfehlern.

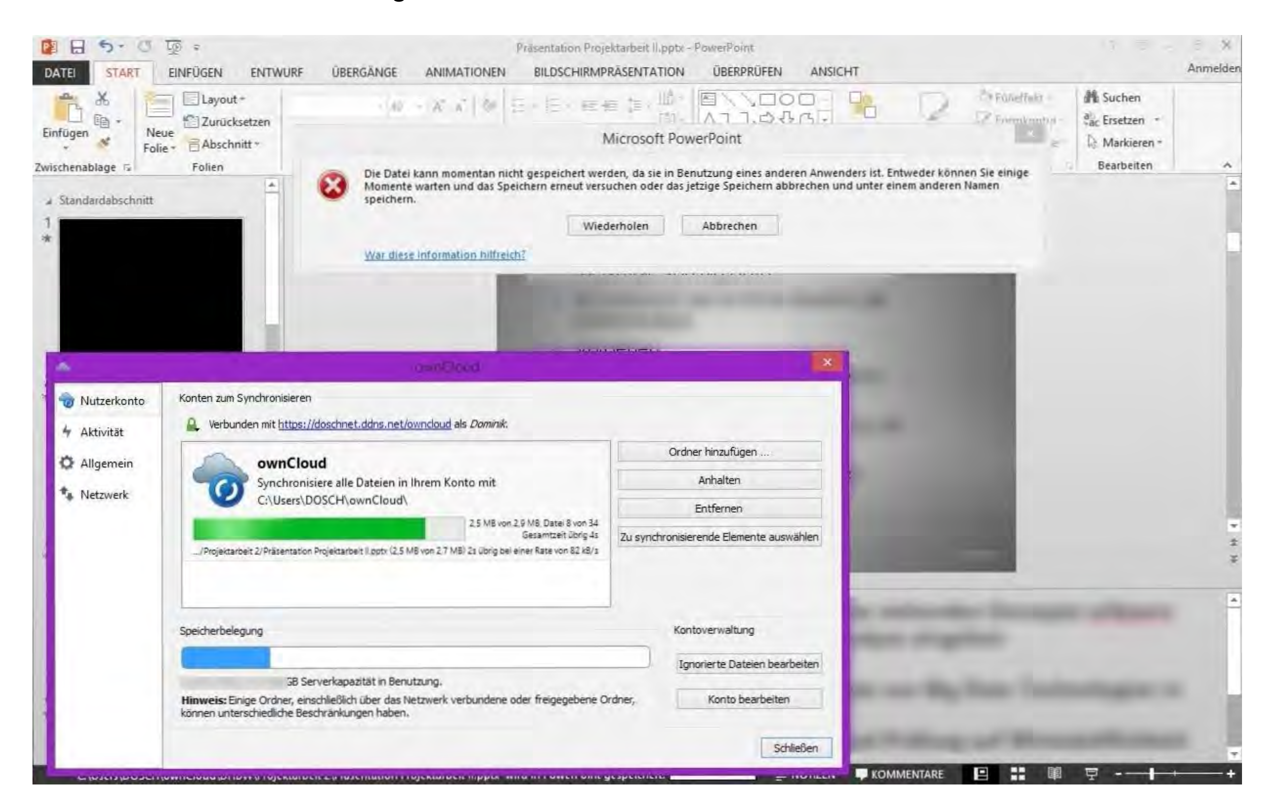


Abb. 28: Fehlermeldung von Microsoft PowerPoint nach dem wiederholten Speichern einer Datei in einen mittels ownCloud-Client synchronisierten Ordners während der Synchronisation der betreffenden Datei

Der Zugriff auf Dateien über die offizielle ownCloud-App funktionierte ohne Probleme. Dagegen verliefen die ersten Tests von WebDAV-Clients negativ. So ergab jeder Synchronisationsversuch mittels FolderSync einen „HTTP 400 Bad Request“. Das selbe Verhalten legte auch ein Dateimanager an den Tag, nachdem ein ownCloud-Konto eingerichtet wurde. Hierbei handelte es sich um einen Bug in der getesteten ownCloud-Version 7.0.3 vom 13.11.2014, der den Zugriff per WebDAV zunächst unmöglich macht. Mit der am 9.12.2014 veröffentlichten Version 7.0.4 ist dieses Problem jedoch behoben worden, worauf der Changelog-Eintrag „fixing port configuration in trusted domains“[6] hinweist. Das Problem ließ sich jedoch bereits vorher durch einen sehr einfachen Workaround lösen, der die bereits im Abschnitt 9.1 Installation und grundlegende Konfiguration erwähnte .htaccess-Datei im ownCloud Stammverzeichnis betraf. Das sich in dieser Datei befindliche „trusted_domains“-Array, in dem vertrauenswürdige Zugriffsadressen definiert sind, muss für jede Zugriffsdomain um einen Eintrag nach dem Schema „[Domain]:443“, für den verwendeten SSL-Port, ergänzt werden. Das Array

```
'trusted-domains' => array (

0 => 'example.com',),
```

, dass den ownCloud-Zugriff nicht über die IP-Adresse des Servers sondern ausschließlich über eine Beispieldomain erlaubt, muss in

```
'trusted-domains' => array (

0 => 'example.com',

1 => 'example.com:443',),
```

geändert werden. Nach dieser Anpassung funktionierte die Synchronisation per WebDAV auch mit der ownCloud Version 7.0.3 zuverlässig. Dabei fiel auf, dass die Synchronisationsdauer unter FolderSync mit der Anzahl der Dateien, unabhängig von deren Größe, deutlich zunimmt. Dies gilt auch dann, wenn keine Änderungen vorgenommen wurden und somit keine Dateien synchronisiert werden müssen. Möglicherweise ist dieses Verhalten jedoch auf den verhältnismäßig leistungsschwachen, im Test verwendeten, Server zurückzuführen. Diese Vermutung wird durch die Beobachtung unterstützt, dass insbesondere die CPU-Auslastung des Servers bei der WebDAV-Synchronisation erheblich anstieg.

Neben der Dateisynchronisation erfolgte ein Test der Kontakteverwaltungs-Funktion. Die Migration der Kontakte wurde auf zwei Wegen getestet, die beide problemlos verliefen. Einerseits erfolgte der Import per Drag&Drop aus einem bestehenden Adressbuch in das, wie im Abschnitt Synchronisation mit Desktop-PCs aus dem Kapitel 9.4 Funktionen von ownCloud im produktiven Einsatz beschrieben, angelegte ownCloud-Adressbuch in Thunderbird. Darüber hinaus wurde ein Import von Kontakten aus dem standardisierten Austauschformat VCF über das Webinterface durchgeführt.

Auch die Kalenderfunktion wurde auf verschiedenen Plattformen getestet. Hierbei stellte sich heraus, dass die Einstellung der Zeitzone für die korrekte Eintragung von Terminen sehr zentral ist. Auf allen Clients erfolgt die Erstellung von Terminen auf Basis der eingestellten Zeitzone des Geräts. Der Browser bekommt zunächst keinen Zugriff auf diese und fragt daher beim ersten Aufruf des Kalenders nach einem Standortzugriff. Zwar ist es möglich dies nicht zu erlauben, dann sollte jedoch im Kalender nach Klick auf das Zahnradsymbol die Zeitzone manuell eingestellt werden. Ansonsten werden Termine auf verschiedenen Geräten mit unterschiedlichen Uhrzeiten angezeigt.

Bei längerer Benutzung fiel darüber hinaus auf, dass nach jedem Update der ownCloud-version auf dem Server die maximale Uploadgröße von Dateien auf 513MB zurückgesetzt wurde. Offensichtlich wird dieser, in der .htaccess-Datei im ownCloud Stammverzeichnis,

standardmäßig verwendete Wert beim monatlichen Update jeweils überschrieben. Das manuelle einstellen des Wertes mit den vorgeschlagenen 8000MB stellt zwar keinen großen Aufwand dar, wäre jedoch durchaus vermeidbar.

Alles in allem kann die Community-Version von ownCloud nach umfangreichem Test dennoch als ausgereifte Lösung betrachtet werden. Zwar hat der Test der aktuellen Version einen Bug im WebDAV-Modul aufgedeckt, jedoch ließ sich dieser durch einen Workaround leicht umgehen. An anderer Stelle, wie der Dateisynchronisation, gibt es Verbesserungspotential im Bereich der Performance und Effizienz. Dies fällt insbesondere im Vergleich zu gängigen Public Cloud-Produkten wie Dropbox auf. Dennoch ist festzuhalten, dass die Zuverlässigkeit auf sehr hohem Niveau liegt und daher ein effektiver Einsatz im Unternehmensbereich grundsätzlich möglich ist. Allerdings bietet ownCloud nicht die Flexibilität, die andere Private Cloud-Produkte wie Eucalyptus oder OpenStack mit ihren diversen Modulen bieten. Sofern sich die Anforderungen jedoch ohnehin auf die Speicherung und Synchronisation von Daten zwischen diversen Geräten, sowie die Kontakte- und Kalendersynchronisation beschränken, sollte ownCloud als Lösung in Betracht gezogen werden.

Auch Funktionen zur gemeinsamen Bearbeitung von Dokumenten können im Unternehmenseinsatz einen deutlichen Mehrwert bieten. Diese Funktionen lassen sich zwar höchstwahrscheinlich günstiger mittels eines Public Cloud Produktes realisieren, wie jedoch im Verlauf dieser Arbeit deutlich geworden ist, stellt dieses Vorgehen in vielen Fällen keine Option für die sensiblen Daten einer Versicherung dar. Sofern die beschriebenen Funktionen die Anforderungen decken, lassen sich diese folglich am schnellsten und kostengünstigsten mittels ownCloud umsetzen. Dies zeigte auch die praktische Realisierung im Rahmen dieser Ausarbeitung.

Allerdings ist fraglich, ob der Funktionsumfang tatsächlich einer größeren Zahl von Unternehmen ausreicht. So werden Kalender und Kontakte in der Praxis häufig bereits über Microsofts ActiveSync übertragen. Zur zuverlässigen Datensynchronisation bieten aktuelle Windows Server-Versionen in Zusammenarbeit mit Windows-Betriebssystemen ebenfalls Möglichkeiten, die im Unternehmenseinsatz häufig verwendet werden. Dennoch kann der Einsatz von ownCloud aus mehreren Gründen sinnvoll sein. Einerseits ist die Anbindung mobiler Endgeräte durch offene Schnittstellen wie WebDAV komfortabler möglich, wie auch im Abschnitt Synchronisation mit Smartphones und Tablets des Kapitels 9.4 Funktionen von ownCloud im produktiven Einsatz deutlich wird. Daneben können Außendienstmitarbeitern, die von verschiedenen Geräten auf ihre Dateien und Kalender zugreifen müssen, schneller derartige Dienste bereitgestellt werden. Ein weiterer sehr zentraler Grund zum Einsatz von ownCloud können die Kollaborationsfunktionen sein. So lässt sich eine gemeinsame Bearbeitung von Dokumenten mit den anderen untersuchten Private Cloud-Produkten nicht ohne Umwege realisieren.

Um die Unterschiede genauer untersuchen zu können wäre ein vergleichender Test mit den im Wettbewerb stehenden Produkten an dieser Stelle sinnvoll gewesen. Die begrenzte Zeit ließ jedoch lediglich die Betrachtung von ownCloud zu, das auch ohne komplexere Infrastruktur-Anforderungen, wie mehreren Rechnerknoten und Hypervisoren auskommt.

9.6 Unterschiede der Enterprise Edition

Wie bereits einleitend in dieses Kapitel erwähnt steht die Enterprise Edition von ownCloud lediglich unter einer kommerziellen Lizenz zur Verfügung. Die Preise hierfür beginnen ab 7200€ für 50 Benutzer.[264] Dieser Abschnitt soll Aufschluss darüber liefern, ob der Einsatz trotz der Kosten im Vergleich zur Community Edition lohnenswert sein kann.

Tests können mit einer 30 Tage lauffähigen Trial-Version durchgeführt werden. Hierauf wurde im Rahmen dieser Ausarbeitung aufgrund der fehlenden Open Source-Lizenz jedoch verzichtet.

Ein großer Teil der Features von ownCloud steht sowohl in der Community- als auch der Enterprise-Edition zur Verfügung. Dennoch sind einige Funktionen ausschließlich der Community Edition vorbehalten. Hierzu gehören beispielsweise die Kontakte-, Kalender- und Foto-App, die Bilder direkt im Webinterface darstellt. Teilweise sind diese im Unternehmenseinsatz jedoch gar nicht notwendig. So legt die in der Enterprise-Edition vorhandene SharePoint-Integration[265] beispielsweise die Kalenderverwaltung über diese Anwendung nahe. Zu beachten ist, dass hierdurch weitere Lizenzkosten anfallen, sofern SharePoint bisher noch nicht verwendet wird. Wenn SharePoint jedoch bereits als zentrales System eingesetzt wird, kann die Anbindung von ownCloud einen erheblichen Mehrwert darstellen. Auch die lediglich in der Enterprise Edition enthaltene SAML/Shibboleth-Unterstützung[266] ist insbesondere für Unternehmen, die dieses Verfahren bereits vor der Einführung von ownCloud zum Single Sign-on verwenden, sinnvoll. Nachdem im professionellen Einsatz häufig Datenbank Management-Systeme von Herstellern wie Oracle oder Microsoft verwendet werden, sind besonders in diesem Bereich umfangreiche Erfahrungen vorhanden. Die Verwendung von MySQL als zusätzlichem Datenbank Management-System ist zwar möglich, von vielen Unternehmen werden jedoch aus diversen Gründen die erstgenannten Systeme bevorzugt. Die Unterstützung für diese stellt ebenfalls ein Alleinstellungsmerkmal der Enter-

[264] Vgl. ownCloud (2014b)
[265] Microsoft SharePoint stellt eine webbasierte Anwendung dar, die zentrale Kollaborationsfunktionen im Unternehmen bereitstellt. Hierzu sind das Dokumentenmanagement, firmeninterne soziale Netzwerke sowie die Projektverwaltung zu zählen.
[266] Shibboleth basiert auf der Security Assertion Markup Language (SAML) und stellt ein standardisiertes Verfahren zur Authentifizierung und Autorisierung gegenüber Webanwendungen dar. Hierbei handelt es sich um eine Single Sign-on-Lösung, die es ermöglicht, dass Benutzer sich nicht verschiedene Zugangsdaten merken müssen, sondern nach einem einfachen Login gegenüber allen Systemen authentifiziert werden.

prise-Edition dar. Darüber hinaus wird ein erweitertes Logging und Reporting angeboten, das für den professionellen Einsatz ebenso wichtig sein kann wie der Support. Letzterer kann in der Enterprise-Edition 24x7 gebucht werden.

Auch im Bereich der Sicherheit bietet die Enterprise-Edition einige sinnvolle Möglichkeiten. So lässt sich über eine Firewall der Dateizugriff nach IP-Adressen, Geräten oder geographischen Zonen regeln. Auf diese Weise kann verhindert werden, dass ein verlorengegangener Link zu einer geteilten, geschäftskritischen Datei, Dritten Zugriff auf diese gibt. Stattdessen kann der Zugriff auf die bekannten IP-Adressen von Außenstellen sowie zugriffsberechtigten Zulieferern oder Kunden beschränkt werden. Abschließend sei noch am Rande erwähnt, dass die Enterprise Edition als primären Speicher zur Datenablage neben den bereits im Kapitel 9.2 Konfiguration über das Webinterface erwähnten Technologien zusätzlich Amazons Simple Storage Service verwenden kann. Dies führt jedoch die grundlegende Idee der Private Cloud ad absurdum.[267]

Zusammenfassend lässt sich feststellen, dass die Enterprise Edition bei der Realisierung einer ownCloud im Unternehmen durchaus in Betracht gezogen werden sollte. So kann insbesondere der Support ein wichtiges Argument darstellen, der die Lizenzkosten bereits rechtfertigt. Auch die Anbindung zentraler Systeme im Unternehmen wie SharePoint oder die Nutzung von bestehenden Datenbank Management-Systemen kann ein wichtiges Argument darstellen. Der Source-Code der Enterprise Edition steht zwar nicht offen zur Verfügung, auf der Download-Site wird jedoch diesbezüglich auf ein Kontakt-Formular verwiesen.[268]

10 Fazit und Ausblick

Autoren: Amelie Böer, Thomas Enderle, Matthias Holzwarth, Dominik Schrahe

Nachdem zu Beginn der Arbeit theoretische Grundlagen beleuchtet wurden, wurde schnell klar, dass in Unternehmen in der Versicherungsbranche vor allem Private Clouds zweckmäßig sind. Den Grund hierfür stellen vor allem der in der Public Cloud nur schwer zu gewährleistende Datenschutz und die Datensicherheit dar. Aus diesem Grund wurden auch im Kriterienkatalog ausschließlich derartige Produkte untersucht. Aufgrund des besonderen Sicherheitsbedürfnisses der betrachteten Branche wurde diesem Thema ein besonderer Stellenwert eingeräumt. So geht die Arbeit einerseits auf Verfahren zur Verschlüsselung und dem sicheren Schlüsselaustausch im Bereich der Transportverschlüsselung ein. Andererseits stellt die Fähigkeit zur Multifaktor Authentifizierung auch ein Kriterium zur Bewertung der untersuchten Produkte dar.

[267] Vgl. ownCloud (2014b)
[268] Vgl. ownCloud (2014a)

Weitere im Kriterienkatalog untersuchte Merkmale sind die Bereitstellung verschiedener Dienste zur Datenspeicherung und Netzwerkverwaltung. Neben diesen funktionalen Anforderungen werden nicht-funktionale Kriterien wie die Zeitabstände zwischen Releases, die Dokumentation oder die Popularität bewertet. Darüber hinaus beinhaltet der Kriterienkatalog auch die Performance, Wartungsanforderungen oder Look and Feel größtenteils auf Basis verfügbarer Nutzerbewertungen.

Neben einer statischen Gewichtung der Kriterien durch die _Versicherung sowie eine Bewertung für ein kleines Unternehmen kann eine flexible Bewertung mittels eines Microsoft-Excel-Sheets vorgenommen werden. Hierüber ist es für jedes Unternehmen einfach möglich, eine individuelle Gewichtung der untersuchten Kriterien vorzunehmen. Das Sheet gibt jeweils die Überweinstimmung der Anforderungen mit den untersuchten Produkten in Prozent an.

Über die theoretische Bewertung der wichtigsten Open Source Private-Cloud-Produkte hinaus hatte diese Arbeit auch zum Ziel, eine Implementierung eines Produktes vorzunehmen. Auf diese Weise konnten praktische Erfahrungen mit der Community Edition von ownCloud gesammelt werden. Dabei ging es einerseits um die Untersuchung des gewählten Produktes auf alltägliche Anwendbarkeit in Versicherungsunternehmen. Andererseits wurde auch die Installation und Einrichtung unter besonderer Beachtung der Sicherheitserfordernisse beschrieben. Eine Implementierung der anderen Produkte, insbesondere zur Durchführung vergleichender Tests, konnte aus Zeitgründen nicht umgesetzt werden.

Nach Durchführung der Bewertung sowie der Implementierung von ownCloud kommt die Arbeit zu dem Ergebnis, dass besonders zwei Produkte herausstechen. Auf der einen Seite ist hier OpenStack zu nennen, das mit seinen zahlreichen Modulen und der besonders großen Entwicklergemeinde einen sehr hohen Grad an Flexibilität bietet. Hierdurch können verschiedenste Anwendungen realisiert werden, wobei die Weiterentwicklung durch den Status als De-Facto-Standard unter den Private Clouds gesichert sein sollte.

Andererseits stellt ownCloud ein empfehlenswertes Produkt dar, das zum Einsatz kommen kann, sofern die angebotenen Funktionen die Anforderungen decken. Sofern Open Source kein zwingendes Kriterium darstellt, sollte auch der Einsatz der Enterprise Edition in Betracht gezogen werden. Hervorzuheben ist bei ownCloud die einfache Installation, die es auch kleineren Unternehmen ermöglicht, eine Private Cloud zu nutzen.

Letztendlich kann also keine klare Empfehlung für ein Produkt ausgesprochen werden, da die untersuchten Private Clouds zu heterogen sind. Die Entscheidung eines Unternehmens hängt zudem nicht nur von der Branche sondern in besonderem Maße auch von der Größe ab. Zur Erleichterung der Auswahl sollte daher das erwähnte Excel-Sheet verwendet werden. Auf diese Weise lässt sich schnell ein Produkt auswählen, dass den eigenen Anforderungen am besten entspricht.

11 Anhang

Anhangverzeichnis

Anhang 1 Glossar

diskreter Logarithmus	Der diskrete Logarithmus ist die Entsprechung des „normalen“ Logarithmus in zyklischen Gruppen. Es existieren nur sehr aufwändige Verfahren zu Berechnung, wohingegen die diskrete Exponentation relativ einfach zu berechnen ist.
elastisch	„Bezeichnung für die Fähigkeit eines Rechenzentrums, IT- Ressourcen flexibel und in kurzer Zeit bereitzustellen und wieder freigeben zu können.“[269]
elliptische Kurve	Elliptische Kurven sind algebraische Kurven, die spezielle Eigenschaften haben. Aufgrund dieser Eigenschaften eignen sie sich zur Anwendung in der Kryptographie
Offset	Versatz oder Verschiebung
Substitutionsbox	Mithilfe von Substitutionsboxen werden bei Blockverschlüsselungen verwendet, um Werte zu vertauschen. Ein festgelegter Eingabewert entspricht dabei einem festgelegten Ausgabewert. Bei AES werden Bytes vertauscht.

[269] Piekenbrock, D./Fehling, C./Leymann, F. (o. J.)

Anhang 2 Releasefrequenz Eucalyptus

Releasenr.	Releasedatum	Abstand
3.0.0	8.2.2012	
3.0.1	13.03.2012	34
3.1.0	22.6.2012	101
3.1.1	17.09.2012	87
3.1.2	27.09.2012	10
3.2.0	18.12.2012	82
3.2.1	28.02.2013	72
3.2.2	16.04.2013	47
3.3.0	20.06.2013	65
3.3.1	11.09.2013	83
3.3.2	24.10.2013	43
3.4.0	24.10.2013	0
3.4.1	16.12.2013	53
3.4.2	25.02.2014	71
4.0.0	30.03.2014	33
4.0.1	04.08.2014	127
4.0.2	03.11.2014	91
	Durchschnitt:	**57,54**

Anhang 3 Releasefrequenz OpenStack

Releasenr.	Releasedatum	Abstand
2012.1	05.04.2012	
2012.1.1	22.06.2012	78
2012.1.1	10.08.2012	49
2012.2	27.09.2012	48
2012.2.1	29.11.2012	63
2012.2.2	13.12.2012	14
2012.2.3	31.01.2013	49
2013.1	04.04.2013	63
2013.1.1	09.05.2013	35
2013.1.2	06.06.2013	28
2013.1.3	08.08.2013	63
2013.1.4	17.10.2013	70
2013.2	17.10.2013	0
2013.2.1	16.12.2013	60
2013.2.2	13.02.2014	59
2013.2.3	03.04.2014	49
2014.1	17.04.2014	14
2014.1.1	09.06.2014	53
2014.1.2	08.08.2014	60
2014.1.3	02.10.2014	55
2014.2	16.10.2014	14
2014.2.1	05.12.2014	50
	Durchschnitt:	**46,38**

Anhang 4 Releasefrequenz ownCloud

Releasenr.	Releasedatum	Abstand
3.0.1	03.04.2012	
3.0.2	11.04.2012	8
3.0.3	27.04.2012	16
4.0.0	22.05.2012	25
4.0.1	04.06.2012	13
4.0.2	11.06.2012	7
4.0.3	23.06.2012	12
4.0.4	28.06.2012	5
4.0.5	20.07.2012	22
4.0.6	01.08.2012	12
4.5.0	09.10.2012	69
4.5.1	23.10.2012	14
4.5.2	14.11.2012	22
4.5.3	26.11.2012	12
4.5.4	03.12.2012	7
4.5.5	20.12.2012	17
4.5.6	22.01.2013	33
4.5.7	19.02.2013	28
4.5.8	14.03.2013	23
5.0.0	14.03.2013	0
5.0.1	02.04.2013	19
5.0.2	02.04.2013	0
5.0.3	03.04.2013	1
5.0.4	11.04.2013	8
5.0.5	19.04.2013	8
5.0.6	14.05.2013	25
5.0.7	06.06.2013	23
5.0.8	09.07.2013	33
5.0.9	15.07.2013	6
5.0.10	12.08.2013	28
5.0.11	10.09.2013	29
5.0.12	03.10.2013	23
5.0.13	05.11.2013	33
6.0.0	11.12.2013	36
6.0.1	22.01.2014	42
6.0.2	03.03.2014	40
6.0.3	29.04.2014	57
6.0.4	23.06.2014	55
7.0.0	23.07.2014	30
7.0.1	04.08.2014	12
7.0.2	26.08.2014	22
7.0.3	22.10.2014	57
7.0.4	09.12.2014	48
	Durchschnitt:	23,33

Anhang 5 Detaillierte Beschreibung des Tools

Allgemeiner Aufbau

Das Tool enthält sechs Tabellenblätter: die Übersicht, Detailansichten für alle vier untersuchten Private Cloud-Produkte und die Bewertungsgrundlage, welche eine Übersicht über alle vergebenen Punkte enthält.

Um zu verhindern, dass aus Versehen eingegebene Daten zu unerlaubten Änderungen und einer hiermit verbundenen Ungültigkeit des Ergebnisses führen, ist das Blattschutz auf allen Tabellenblättern aktiviert. Für die Deaktivierung ist die Eingabe des Passwortes „cloud“ notwendig.

Übersicht

In Abb. 29 ist das Tabellenblatt Übersicht dargestellt. Alle grauen Zellen sind nicht gesperrt – dies ist die einzige Möglichkeit ohne Aufhebung des Blattschutzes etwas einzugeben.

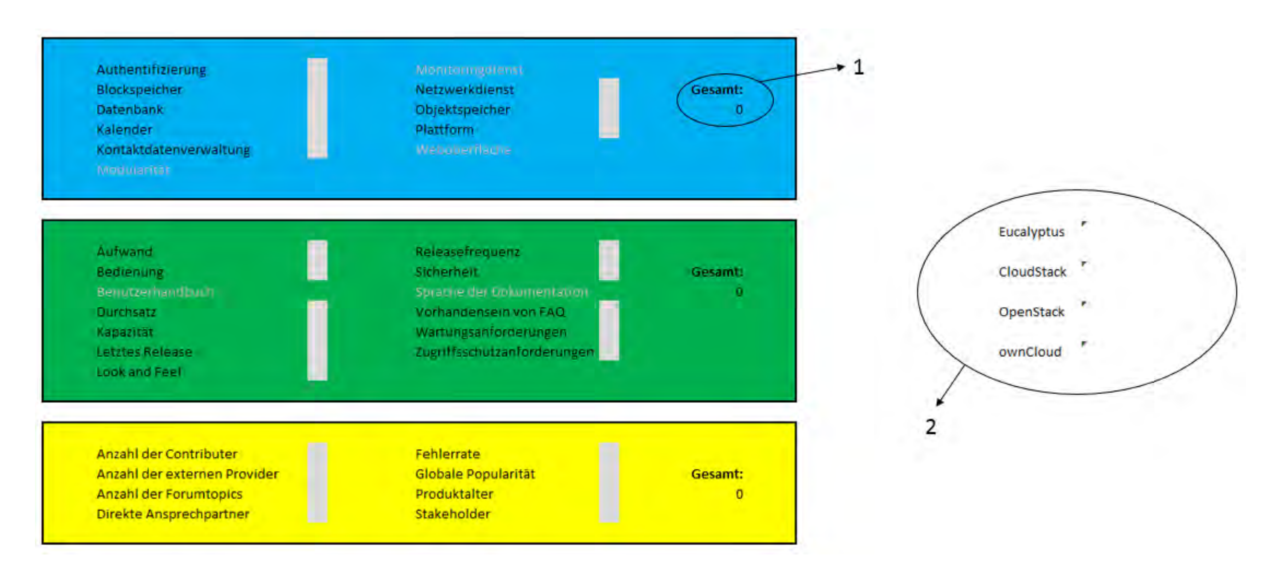


Abb. 29: Tabellenblatt Übersicht – Gesamtansicht

Für den blauen Bereich, die funktionalen Kriterien, ist bereits eine Eingabe ab null möglich. Dies wird über die Datenüberprüfung sichergestellt. Bei Eingabe einer negativen Zahl erscheint die in Abb. 30 dargestellte Warnung.

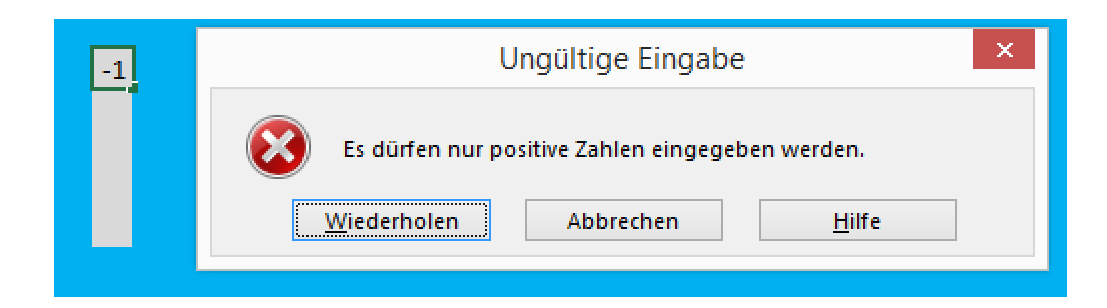


Abb. 30: Tabellenblatt Übersicht – Datenüberprüfung der funktionalen Kriterien

Solange noch Eingaben offen sind, werden diese grau markiert. Dies ist wird über die bedingte Formatierung, vgl. hierzu Abb. 31, ermöglicht.

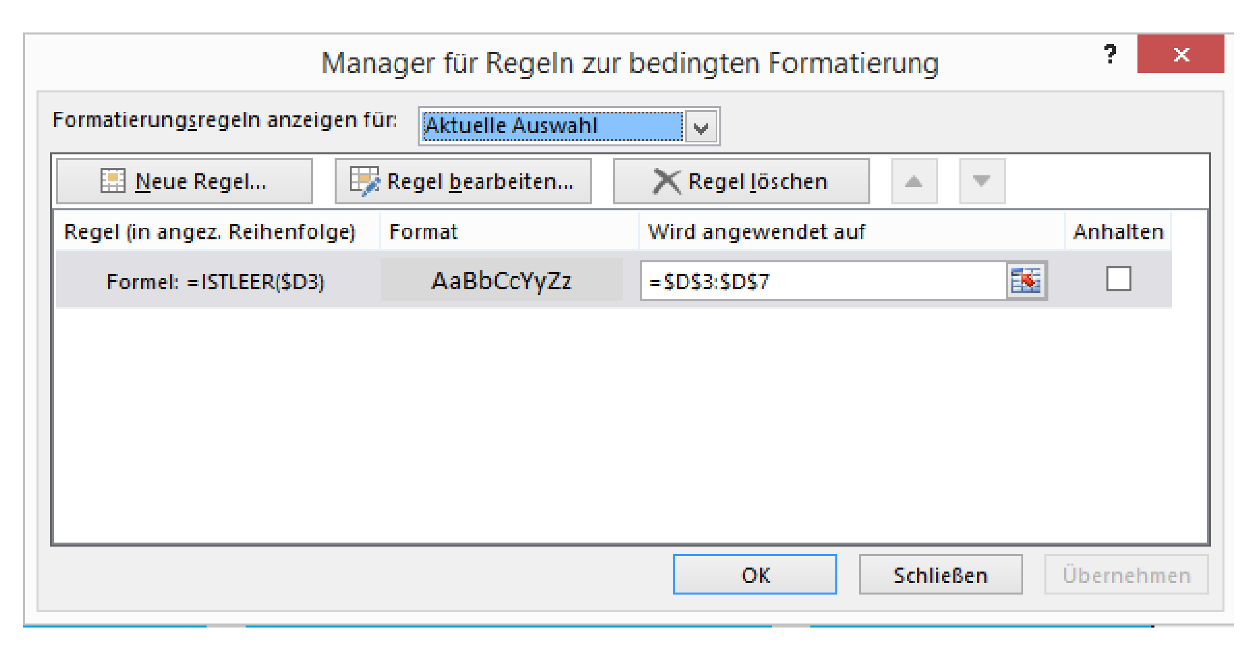


Abb. 31: Tabellenblatt Übersicht – Bedingte Formatierung der funktionalen Kriterien

Die Berechnung der Gesamtgewichtung für die funktionalen Kriterien, vgl. Abb. 29 Ziffer 1 wird über eine Summenfunktion erreicht.

Der grüne und der gelbe Bereich sind gleich formatiert. Die korrekte Eingabe von Gewichtungen wird abermals über die Datenüberprüfung gewährleistet. Diesmal dürfen allerdings keine Zahlen kleiner eins eingegeben werden. Die Markierung der grauen Felder erfolgt erneut über die bedingte Formatierung. Die Zellen sind standardmäßig grau. Bei Eingabe eines Wertes werden sie entsprechend grün bzw. gelb markiert – vgl. hierzu Abb. 32. Die Ermittlung der Gesamtgewichtung läuft parallel zum funktionalen Bereich ab.

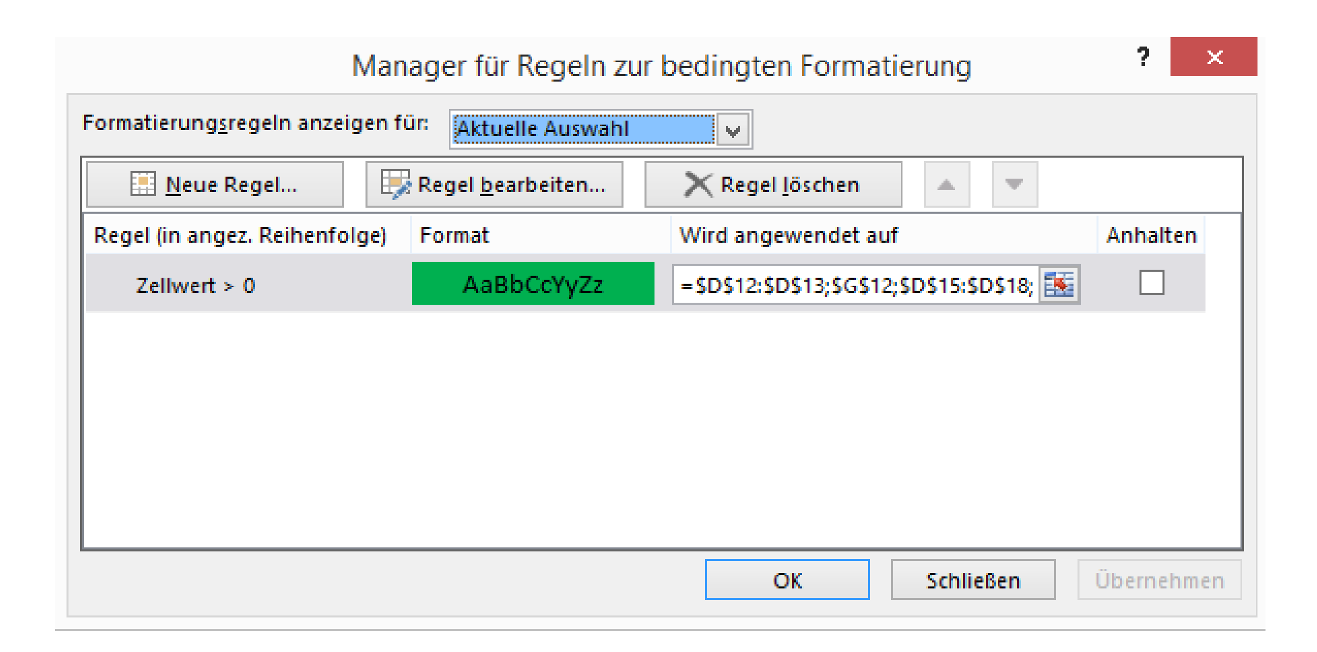


Abb. 32: Tabellenblatt Übersicht – Bedingte Formatierung der nicht-funktionalen und der Open Source-Kriterien

Die Ergebnisse der Clouds, s. Abb. 29 Z. 2, werden aus den Detailansichten, s. Abb. 34 Z. 1, übernommen. Über die bedingte Formatierung werden die Werte entsprechend einer abgestuften Farbskala der Größe nach markiert. Solange noch keine Gewichtung eingefügt ist, entsteht aufgrund einer Division durch Null ein Fehler in den Detailansichten. Die Anzeige von diesem in der Übersicht verhindert abermals eine bedingte Formatierung.

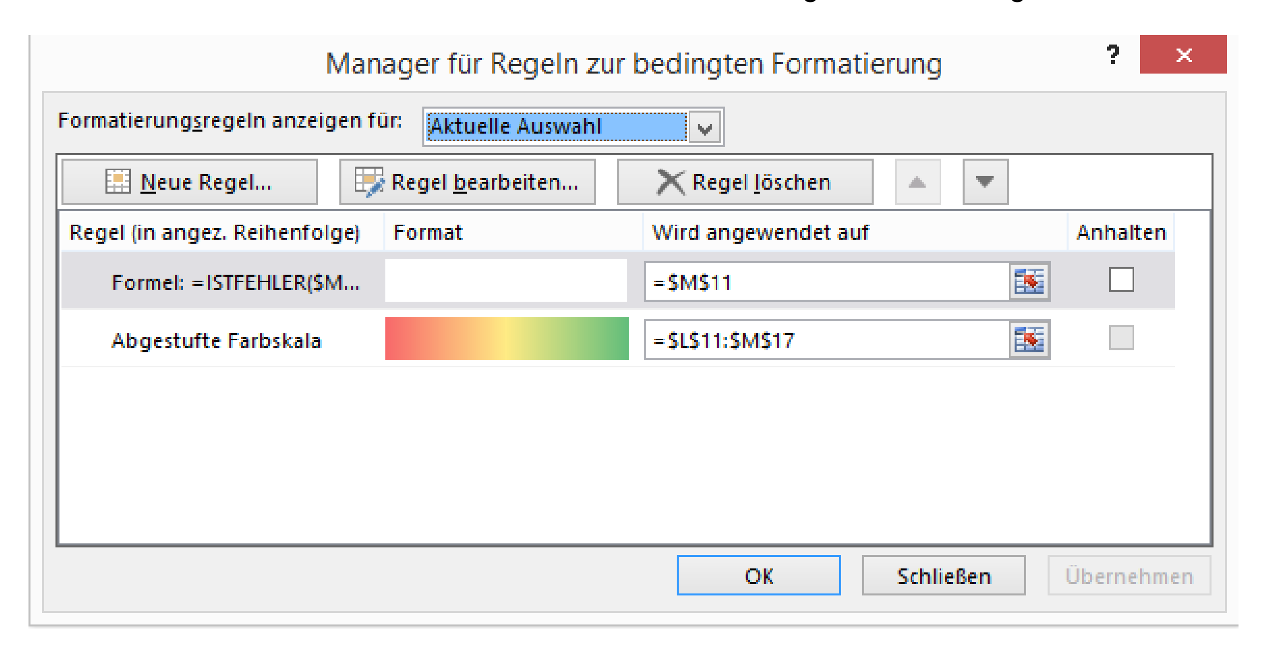


Abb. 33: Tabellenblatt Übersicht – Bedingte Formatierung des Ergebnisses

Alle Kriterien, bei denen es keine Unterschiede zwischen den Produkten gab, sind in grauer Schrift dargestellt. Hier kann keine Eingabe einer Gewichtung erfolgen. Dies wurde manuell, also ohne Benutzung einer Formel, eingestellt.

Detailansichten

Der Aufbau der Detailansichten ist für alle Produkte gleich. Er wird hier beispielhaft für das Produkt Eucalyptus aufgezeigt. In Abb. 34 ist das gesamte Tabellenblatt dargestellt. Das Tabellenblatt ist vollständig geschützt – hier sind keine Eingaben möglich. Durch Fixierung des Tabellenblattes im oberen Bereich wird die horizontale Tabellenbeschriftung auch bei Scrollen weiterhin angezeigt.

Eucalyptus		2	3	4	5
	Kriterium	Punktzahl	Gewichtung	Kriterien-punktzahl	Maximal zu erreichende Punktzahl
Funktionale Kriterien	Authentifizierung	1	0	0	0
	Blockspeicher	2	0	0	0
	Datenbank	2	0	0	0
	Kalender	0	0	0	0
	Kontaktdatenverwaltung	0	0	0	0
	Modularität	2	0	0	0
	Monitoringdienst	2	0	0	0
	Netzwerkdienst	2	0	0	0
	Objektspeicher	2	0	0	0
	Plattform	1	0	0	0
	Weboberfläche	2	0	0	0
Nicht-funktionale Kriterien	Aufwand	1	0	0	0
	Bedienung	1	0	0	0
	Benutzerhandbuch	2	0	0	0
	Durchsatz	0	0	0	0
	Kapazität	1	0	0	0
	Letztes Release	1	0	0	0
	Look and Feel	0	0	0	0
	Releasefrequenz	1	0	0	0
	Sicherheit	0	0	0	0
	Sprache der Dokumentation	1	0	0	0
	Vorhandensein von FAQ	0	0	0	0
	Wartungsanforderungen	1	0	0	0
	Zugriffsschutzanforderungen	2	0	0	0
Open Source Kriterien	Anzahl der Contributer	0	0	0	0
	Anzahl der Externen Provider	2	0	0	0
	Anzahl der Forumtopics	0	0	0	0
	Direkte Ansprechpartner	2	0	0	0
	Fehlerrate	1	0	0	0
	Globale Populariät (Traffic Rank)	0	0	0	0
	Produktalter	2	0	0	0
	Stakeholder	2	0	0	0

6: Erreichte Punktzahl	Maximal erreichbare Punktzahl	1: Ergebnis
0	0	

Abb. 34: Tabellenblatt Eucalyptus – Gesamtansicht

Die Punktezahl des jeweiligen Kriteriums, vgl. Abb. 34 Z. 2, wird von dem Tabellenblatt Bewertungsgrundlage übernommen. Die Gewichtung, vgl. Abb. 34 Z. 3, ist mit der jeweiligen Gewichtung der Übersicht verknüpft. Nur in Einzelfällen wird diese manuell auf null gesetzt. Dies ist bei der Bewertung des jeweiligen Produktes in der Ausarbeitung ersichtlich. Die Kriterienpunktzahl, Abb. 34 Z. 4, errechnet sich aus einer Multiplikation von der Punktzahl mit der Gewichtung. Die maximal zu erreichende Punktzahl, Abb. 34 Z. 5, aus der Multiplikation der Gewichtung mit zwei. Alle Zeilen, welche mit einer Gewichtung von über vier versehen

sind, werden gelb hervorgehoben. Dies wird über eine bedingte Formatierung, s. Abb. 35, ermöglicht.

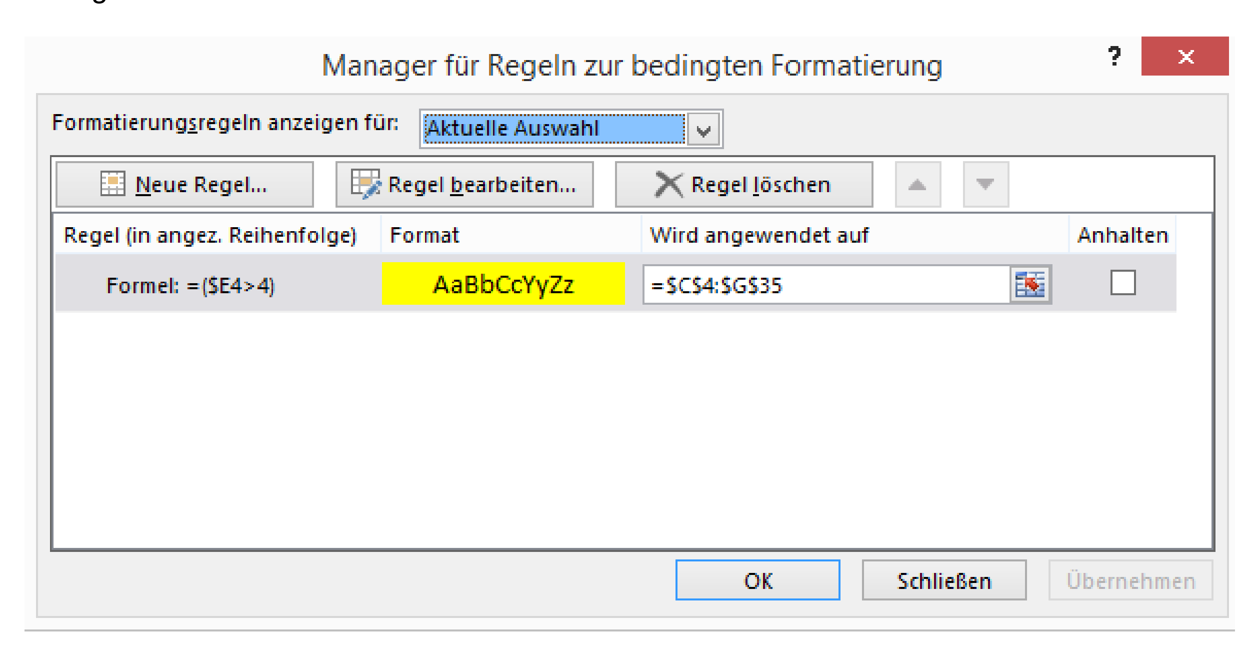


Abb. 35: Tabellenblatt Eucalyptus – Bedingte Formatierung

Die erreichte Punktzahl und die gesamt maximal erreichbare Punktzahl, s. Abb. 34 Z. 6, werden über Summenfunktionen ermittelt. Das Ergebnis, s. Abb. 34 Z. 1, errechnet sich aus einer Division der erreichten Punktzahl durch die maximal mögliche und ist als Prozentzahl formatiert.

Bewertungsgrundlage

Die Bewertungsgrundlage, abgebildet in Abb. 36 ist komplett statisch erstellt. Hier kommen keine Formeln, bedingte Formatierungen oder ähnliches zum Einsatz. Die horizontale Tabellenbeschriftung wird durch eine Fixierung erneut auch bei Scrollen weiterhin angezeigt.

	Eucalyptus	Cloud Stack	OpenStack	ownCloud
Authentifizierung	1	1	2	2
Blockspeicher	2	2	2	0
Datenbank	2	0	2	0
Kalender	0	0	0	2
Kontaktdatenverwaltung	0	0	0	2
Modularität	2	2	2	2
Monitoringdienst	2	2	2	2
Netzwerkdienst	2	2	2	0
Objektspeicher	2	2	2	0
Plattform	1	1	1	2
Weboberfläche	2	2	2	2
Aufwand	1	1	0	2
Bedienung	1	1	0	2
Benutzerhandbuch	2	2	2	2
Durchsatz	0	1	1	2
Kapazität	1	0	1	2
Letztes Release	1	1	1	2
Look and Feel	0	1	1	2
Releasefrequenz	1	0	1	2
Sicherheit	0	2	1	1
Sprache der Dokumentation	1	1	1	1
Vorhandensein von FAQ	0	2	2	2
Wartungsanforderungen	1	0	1	2
Zugriffsschutzanforderungen	2	1	0	1
Anzahl der Contributer	0	1	2	2
Anzahl der Externen Provider	2	2	2	1
Anzahl der Forumtopics	0	0	1	1
Direkte Ansprechpartner	2	0	0	2
Fehlerrate	1	2	1	1
Globale Popularität (Traffic Rank)	0		1	2
Produktalter	2	1	1	1
Stakeholder	2	2	2	0

Abb. 36: Bewertungsgrundlage – Gesamtansicht

Anhang 6 Übersicht der Gewichtungsempfehlung

	Kriterium	1	2	3	4	5
Funktionale Kriterien	**Aufwand**	Mittlere bis große Unternehmen			Kleines Unternehmen	
	Bedienung				x	
	Durchsatz	Individuell abhängig vom geplanten Einsatz				
	Kapazität	Individuell abhängig vom geplanten Einsatz				
	Letztes Release	X				
	Look and Feel			Einführung aufgrund Mitarbeiterwünschen	x	IT-fremde Benutzer
	Performance	Individuell abhängig vom geplanten Einsatz				
	Releasefrequenz				x	
	Vorhandensein von FAQ	Bereits Erfahrung vorhanden	x	Kein IT-Spezialist vorhanden		
	Wartungs-anforderungen	Aufwand + 1				
	Zugriffsschutz-anforderungen				x	
Open Source Kriterien	**Anzahl der Forumtopics**				x	
	Anzahl der Contributer				x	
	Anzahl externer Provider	Kein Provider geplant	x	Provider optional		Provider erforderlich
	Direkte Ansprechpartner		x			
	Fehlerrate	x				
	Globale Popularität		x			
	Produktalter			x		
	Stakeholder		Große Unternehmen		Kleine bis mittlere Unternehmen	

Anhang 7 Detaillierte Ergebnisse _Versicherung

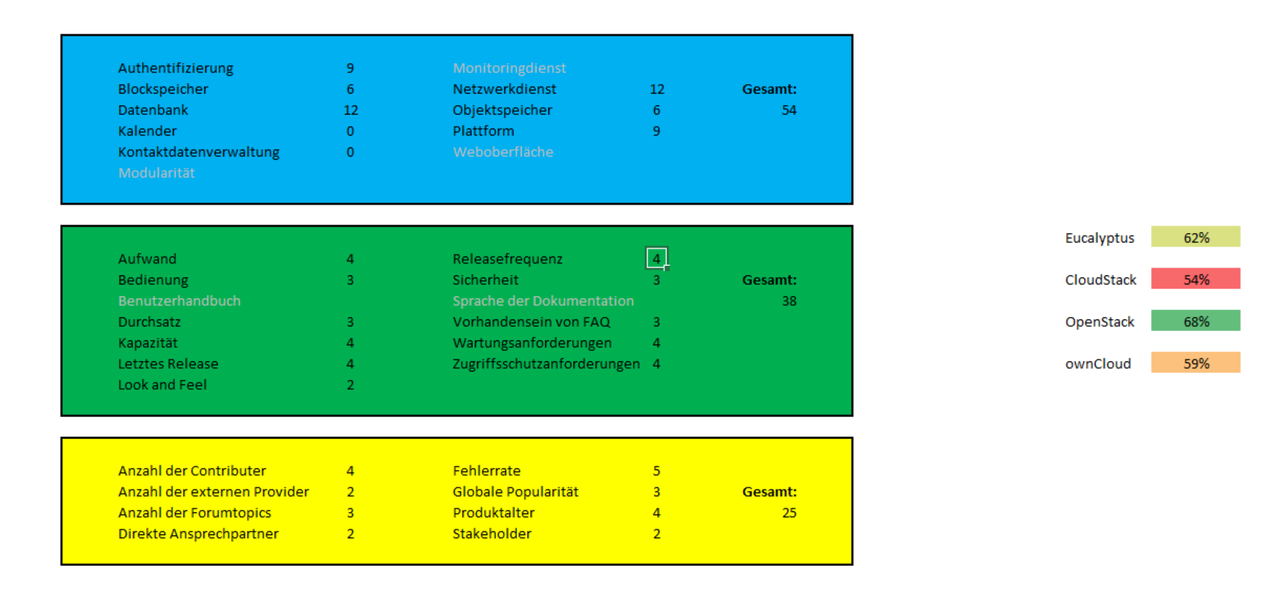


Abb. 37: _Versicherung – Gesamtübersicht

Eucalyptus

	Kriterium	Punktzahl	Gewichtung	Kriterien-punktzahl	Maximal zu erreichende Punktzahl
Funktionale Kriterien	Authentifizierung	1	9	9	18
	Blockspeicher	2	6	12	12
	Datenbank	2	12	24	24
	Kalender	0	0	0	0
	Kontaktdatenverwaltung	0	0	0	0
	Modularität	2	0	0	0
	Monitoringdienst	2	0	0	0
	Netzwerkdienst	2	12	24	24
	Objektspeicher	2	6	12	12
	Plattform	1	9	9	18
	Weboberfläche	2	0	0	0
Nicht-funktionale Kriterien	Aufwand	1	4	4	8
	Bedienung	1	3	3	6
	Benutzerhandbuch	2	0	0	0
	Durchsatz	0	3	0	6
	Kapazität	1	4	4	8
	Letztes Release	1	4	4	8
	Look and Feel	0	2	0	4
	Releasefrequenz	1	4	4	8
	Sicherheit	0	3	0	6
	Sprache der Dokumentation	1	0	0	0
	Vorhandensein von FAQ	0	3	0	6
	Wartungsanforderungen	1	4	4	8
	Zugriffsschutzanforderungen	2	4	8	8
Open Source Kriterien	Anzahl der Contributer	0	4	0	8
	Anzahl der Externen Provider	2	2	4	4
	Anzahl der Forumtopics	0	3	0	6
	Direkte Ansprechpartner	2	2	4	4
	Fehlerrate	1	5	5	10
	Globale Populariät (Traffic Rank)	0	3	0	6
	Produktalter	2	4	8	8
	Stakeholder	2	2	4	4

Erreichte Punktzahl	Maximal erreichbare Punktzahl	**Ergebnis**
146	234	**62%**

Abb. 38: _Versicherung – Detailansicht Eucalyptus

CloudStack					
	Kriterium	Punktzahl	Gewichtung	Kriterien-punktzahl	Maximal zu erreichende Punktzahl
Funktionale Kriterien	Authentifizierung	1	9	9	18
	Blockspeicher	2	6	12	12
	Datenbank	0	12	0	24
	Kalender	0	0	0	0
	Kontaktdatenverwaltung	0	0	0	0
	Modularität	2	0	0	0
	Monitoringdienst	2	0	0	0
	Netzwerkdienst	2	12	24	24
	Objektspeicher	2	6	12	12
	Plattform	1	9	9	18
	Weboberfläche	2	0	0	0
Nicht-funktionale Kriterien	Aufwand	1	4	4	8
	Bedienung	1	3	3	6
	Benutzerhandbuch	2	0	0	0
	Durchsatz	1	3	3	6
	Kapazität	0	4	0	8
	Letztes Release	1	4	4	8
	Look and Feel	1	2	2	4
	Releasefrequenz	0	4	0	8
	Sicherheit	2	3	6	6
	Sprache der Dokumentation	1	0	0	0
	Vorhandensein von FAQ	2	3	6	6
	Wartungsanforderungen	0	4	0	8
	Zugriffsanforderungen	1	4	4	8
Open Source Kriterien	Anzahl der Contributer	1	4	4	8
	Anzahl der Externen Provider	2	2	4	4
	Anzahl der Forumtopics	0	3	0	6
	Direkte Ansprechpartner	0	2	0	4
	Fehlerrate	2	5	10	10
	Globale Populariät (Traffic Rank)	0	0	0	0
	Produktalter	1	4	4	8
	Stakeholder	2	2	4	4

Erreichte Punktzahl	Maximal erreichbare Punktzahl	**Ergebnis**
124	228	**54%**

Abb. 39: _Versicherung – Detailansicht CloudStack

OpenStack					
	Kriterium	Punktzahl	Gewichtung	Kriterien-punktzahl	Maximal zu erreichende Punktzahl
Funktionale Kriterien	Authentifizierung	2	9	18	18
	Blockspeicher	2	6	12	12
	Datenbank	2	12	24	24
	Kalender	0	0	0	0
	Kontaktdatenverwaltung	0	0	0	0
	Modularität	2	0	0	0
	Monitoringdienst	2	0	0	0
	Netzwerkdienst	2	12	24	24
	Objektspeicher	2	6	12	12
	Plattform	1	9	9	18
	Weboberfläche	2	0	0	0
Nicht-funktionale Kriterien	Aufwand	0	4	0	8
	Bedienung	0	3	0	6
	Benutzerhandbuch	2	0	0	0
	Durchsatz	1	3	3	6
	Kapazität	1	4	4	8
	Letztes Release	1	4	4	8
	Look and Feel	1	2	2	4
	Releasefrequenz	1	4	4	8
	Sicherheit	1	3	3	6
	Sprache der Dokumentation	1	0	0	0
	Vorhandensein von FAQ	2	3	6	6
	Wartungsanforderungen	1	4	4	8
	Zugriffsanforderungen	0	4	0	8
Open Source Kriterien	Anzahl der Contributer	2	4	8	8
	Anzahl der Externen Provider	2	2	4	4
	Anzahl der Forumtopics	1	3	3	6
	Direkte Ansprechpartner	0	2	0	4
	Fehlerrate	1	5	5	10
	Globale Populariät (Traffic Rank)	1	3	3	6
	Produktalter	1	4	4	8
	Stakeholder	2	2	4	4

Erreichte Punktzahl	Maximal erreichbare Punktzahl	**Ergebnis**
160	234	**68%**

Abb. 40: _Versicherung – Detailansicht OpenStack

ownCloud					
	Kriterium	Punktzahl	Gewichtung	Kriterien-punktzahl	Maximal zu erreichende Punktzahl
Funktionale Kriterien	Authentifizierung	2	9	18	18
	Blockspeicher	0	6	0	12
	Datenbank	0	12	0	24
	Kalender	2	0	0	0
	Kontaktdatenverwaltung	2	0	0	0
	Modularität	2	0	0	0
	Monitoringdienst	2	0	0	0
	Netzwerkdienst	0	12	0	24
	Objektspeicher	0	6	0	12
	Plattform	2	9	18	18
	Weboberfläche	2	0	0	0
Nicht-funktionale Kriterien	Aufwand	2	4	8	8
	Bedienung	2	3	6	6
	Benutzerhandbuch	2	0	0	0
	Durchsatz	2	3	6	6
	Kapazität	2	4	8	8
	Letztes Release	2	4	8	8
	Look and Feel	2	2	4	4
	Releasefrequenz	2	4	8	8
	Sicherheit	1	3	3	6
	Sprache der Dokumentation	1	0	0	0
	Vorhandensein von FAQ	2	3	6	6
	Wartungsanforderungen	2	4	8	8
	Zugriffsanforderungen	1	4	4	8
Open Source Kriterien	Anzahl der Contributer	2	4	8	8
	Anzahl der Externen Provider	1	2	2	4
	Anzahl der Forumtopics	1	3	3	6
	Direkte Ansprechpartner	2	2	4	4
	Fehlerrate	1	5	5	10
	Globale Populariät (Traffic Rank)	2	3	6	6
	Produktalter	1	4	4	8
	Stakeholder	0	2	0	4

Erreichte Punktzahl	Maximal erreichbare Punktzahl	**Ergebnis**
137	234	59%

Abb. 41: _Versicherung – Detailansicht ownCloud

Anhang 8 Detaillierte Ergebnisse Freund

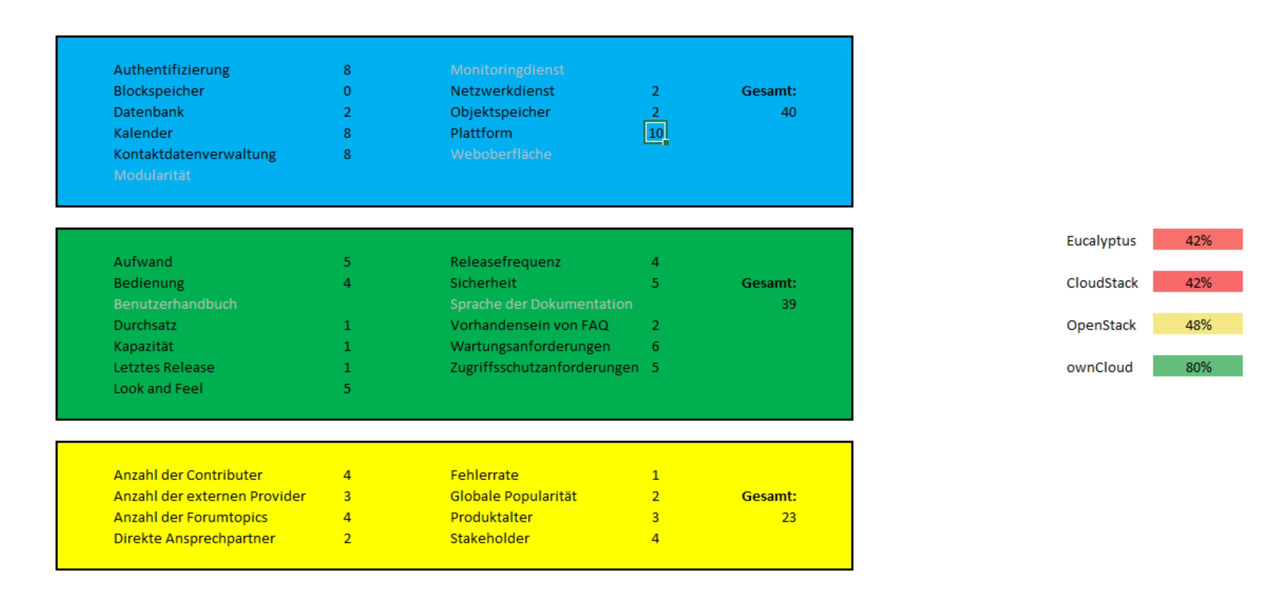


Abb. 42: Freund – Gesamtübersicht

Eucalyptus

	Kriterium	Punktzahl	Gewichtung	Kriterien-punktzahl	Maximal zu erreichende Punktzahl
Funktionale Kriterien	Authentifizierung	1	8	8	16
	Blockspeicher	2	0	0	0
	Datenbank	2	2	4	4
	Kalender	0	8	0	16
	Kontaktdatenverwaltung	0	8	0	16
	Modularität	2	0	0	0
	Monitoringdienst	2	0	0	0
	Netzwerkdienst	2	2	4	4
	Objektspeicher	2	2	4	4
	Plattform	1	10	10	20
	Weboberfläche	2	0	0	0
Nicht-funktionale Kriterien	Aufwand	1	5	5	10
	Bedienung	1	4	4	8
	Benutzerhandbuch	2	0	0	0
	Durchsatz	0	1	0	2
	Kapazität	1	1	1	2
	Letztes Release	1	1	1	2
	Look and Feel	0	5	0	10
	Releasefrequenz	1	4	4	8
	Sicherheit	0	5	0	10
	Sprache der Dokumentation	1	0	0	0
	Vorhandensein von FAQ	0	2	0	4
	Wartungsanforderungen	1	6	6	12
	Zugriffsschutzanforderungen	2	5	10	10
Open Source Kriterien	Anzahl der Contributer	0	4	0	8
	Anzahl der Externen Provider	2	3	6	6
	Anzahl der Forumtopics	0	4	0	8
	Direkte Ansprechpartner	2	2	4	4
	Fehlerrate	1	1	1	2
	Globale Populariät (Traffic Rank)	0	2	0	4
	Produktalter	2	3	6	6
	Stakeholder	2	4	8	8

Erreichte Punktzahl	Maximal erreichbare Punktzahl	**Ergebnis**
86	204	**42%**

Abb. 43: Freund – Detailansicht Eucalyptus

CloudStack						
	Kriterium	Punktzahl	Gewichtung	Kriterien-punktzahl	Maximal zu erreichende Punktzahl	
Funktionale Kriterien	Authentifizierung	1	8	8	16	
	Blockspeicher	2	0	0	0	
	Datenbank	0	2	0	4	
	Kalender	0	8	0	16	
	Kontaktdatenverwaltung	0	8	0	16	
	Modularität	2	0	0	0	
	Monitoringdienst	2	0	0	0	
	Netzwerkdienst	2	2	4	4	
	Objektspeicher	2	2	4	4	
	Plattform	1	10	10	20	
	Weboberfläche	2	0	0	0	
Nicht-funktionale Kriterien	Aufwand	1	5	5	10	
	Bedienung	1	4	4	8	
	Benutzerhandbuch	2	0	0	0	
	Durchsatz	1	1	1	2	
	Kapazität	0	1	0	2	
	Letztes Release	1	1	1	2	
	Look and Feel	1	5	5	10	
	Releasefrequenz	0	4	0	8	
	Sicherheit	2	5	10	10	
	Sprache der Dokumentation	1	0	0	0	
	Vorhandensein von FAQ	2	2	4	4	
	Wartungsanforderungen	0	6	0	12	
	Zugriffsanforderungen	1	5	5	10	
Open Source Kriterien	Anzahl der Contributer	1	4	4	8	
	Anzahl der Externen Provider	2	3	6	6	
	Anzahl der Forumtopics	0	4	0	8	
	Direkte Ansprechpartner	0	2	0	4	
	Fehlerrate	2	1	2	2	
	Globale Populariät (Traffic Rank)	0	0	0	0	
	Produktalter	1	3	3	6	
	Stakeholder	2	4	8	8	
				Erreichte Punktzahl	Maximal erreichbare Punktzahl	**Ergebnis**
				84	200	**42%**

Abb. 44: Freund – Detailansicht CloudStack

OpenStack					
	Kriterium	Punktzahl	Gewichtung	Kriterien-punktzahl	Maximal zu erreichende Punktzahl
Funktionale Kriterien	Authentifizierung	2	8	16	16
	Blockspeicher	2	0	0	0
	Datenbank	2	2	4	4
	Kalender	0	8	0	16
	Kontaktdatenverwaltung	0	8	0	16
	Modularität	2	0	0	0
	Monitoringdienst	2	0	0	0
	Netzwerkdienst	2	2	4	4
	Objektspeicher	2	2	4	4
	Plattform	1	10	10	20
	Weboberfläche	2	0	0	0
Nicht-funktionale Kriterien	Aufwand	0	5	0	10
	Bedienung	0	4	0	8
	Benutzerhandbuch	2	0	0	0
	Durchsatz	1	1	1	2
	Kapazität	1	1	1	2
	Letztes Release	1	1	1	2
	Look and Feel	1	5	5	10
	Releasefrequenz	1	4	4	8
	Sicherheit	1	5	5	10
	Sprache der Dokumentation	1	0	0	0
	Vorhandensein von FAQ	2	2	4	4
	Wartungsanforderungen	1	6	6	12
	Zugriffsanforderungen	0	5	0	10
Open Source Kriterien	Anzahl der Contributer	2	4	8	8
	Anzahl der Externen Provider	2	3	6	6
	Anzahl der Forumtopics	1	4	4	8
	Direkte Ansprechpartner	0	2	0	4
	Fehlerrate	1	1	1	2
	Globale Populariät (Traffic Rank)	1	2	2	4
	Produktalter	1	3	3	6
	Stakeholder	2	4	8	8

Erreichte Punktzahl	Maximal erreichbare Punktzahl	**Ergebnis**
97	204	**48%**

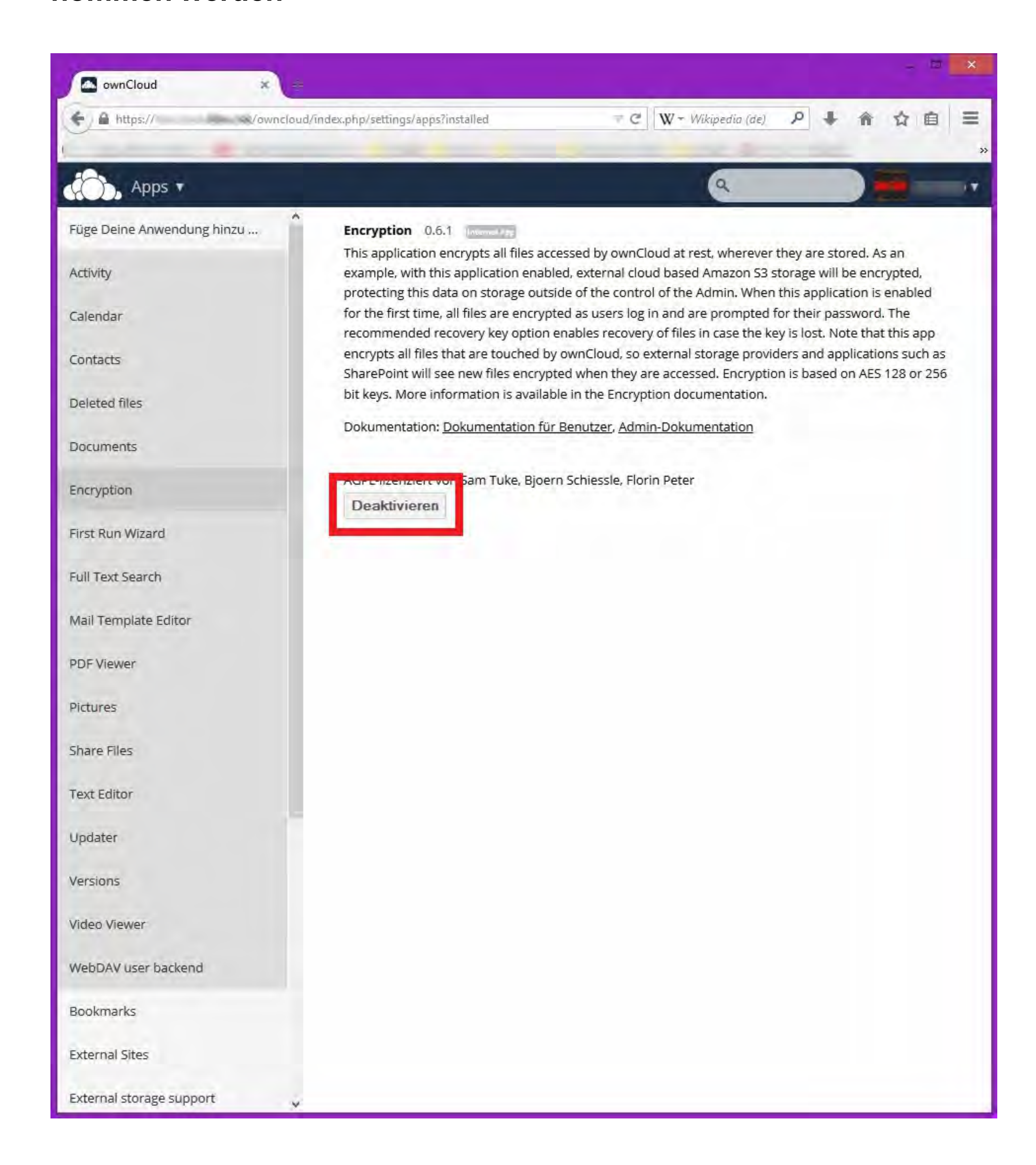
Abb. 45: Freund – Detailansicht OpenStack

ownCloud					
	Kriterium	Punktzahl	Gewichtung	Kriterien-punktzahl	Maximal zu erreichende Punktzahl
Funktionale Kriterien	Authentifizierung	2	8	16	16
	Blockspeicher	0	0	0	0
	Datenbank	0	2	0	4
	Kalender	2	8	16	16
	Kontaktdatenverwaltung	2	8	16	16
	Modularität	2	0	0	0
	Monitoringdienst	2	0	0	0
	Netzwerkdienst	0	2	0	4
	Objektspeicher	0	2	0	4
	Plattform	2	10	20	20
	Weboberfläche	2	0	0	0
Nicht-funktionale Kriterien	Aufwand	2	5	10	10
	Bedienung	2	4	8	8
	Benutzerhandbuch	2	0	0	0
	Durchsatz	2	1	2	2
	Kapazität	2	1	2	2
	Letztes Release	2	1	2	2
	Look and Feel	2	5	10	10
	Releasefrequenz	2	4	8	8
	Sicherheit	1	5	5	10
	Sprache der Dokumentation	1	0	0	0
	Vorhandensein von FAQ	2	2	4	4
	Wartungsanforderungen	2	6	12	12
	Zugriffsanforderungen	1	5	5	10
Open Source Kriterien	Anzahl der Contributer	2	4	8	8
	Anzahl der Externen Provider	1	3	3	6
	Anzahl der Forumtopics	1	4	4	8
	Direkte Ansprechpartner	2	2	4	4
	Fehlerrate	1	1	1	2
	Globale Populariät (Traffic Rank)	2	2	4	4
	Produktalter	1	3	3	6
	Stakeholder	0	4	0	8

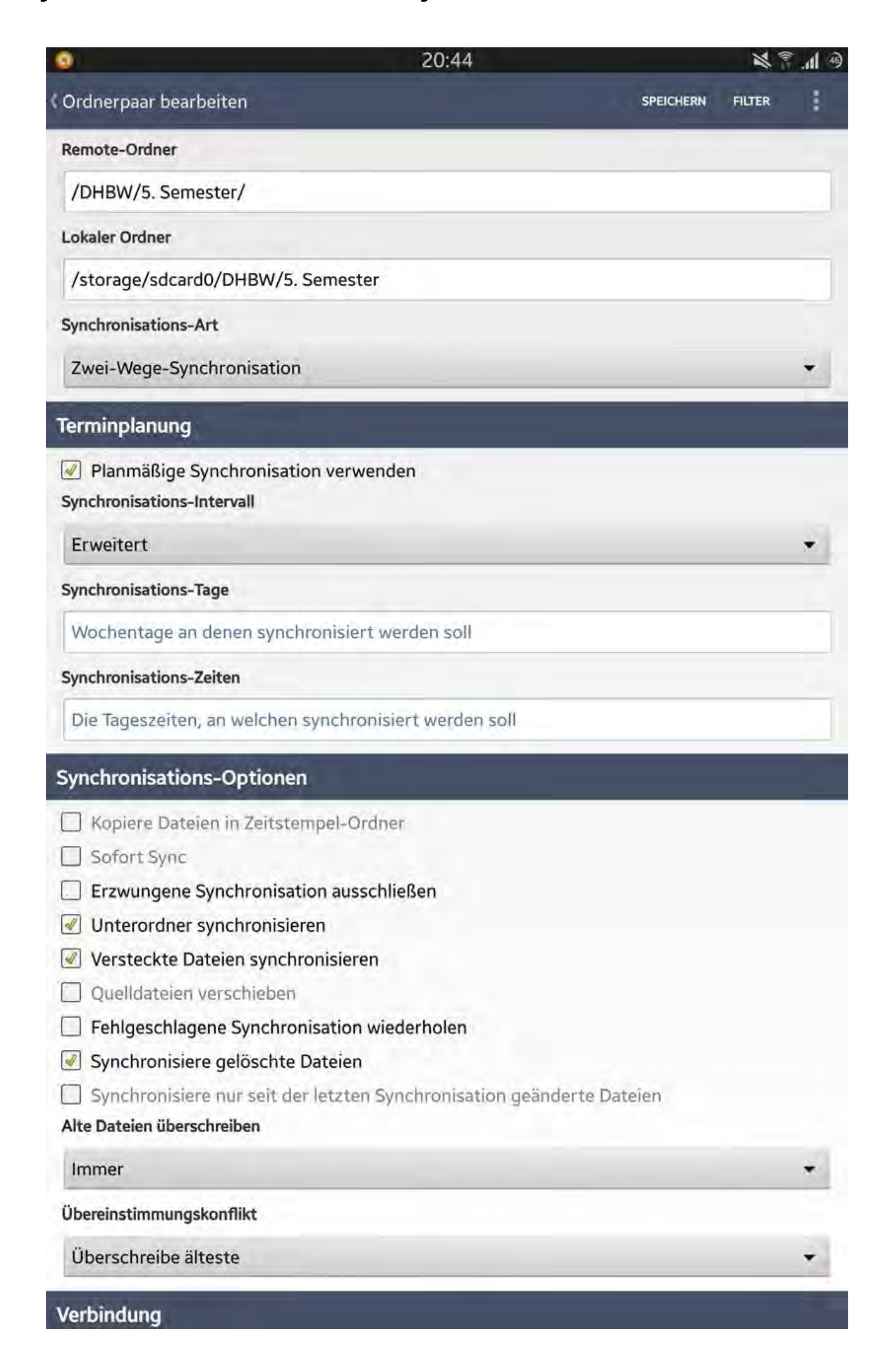
Erreichte Punktzahl	Maximal erreichbare Punktzahl	**Ergebnis**
163	204	**80%**

Abb. 46: Freund – Detailansicht ownCloud

Anhang 9 Das aktivieren und deaktivieren einzelner Apps kann über die entsprechende Schaltfläche in der Appverwaltung vorgenommen werden



Anhang 10 Ausschnitt aus den Einstellungsoptionen zur Verzeichnissynchronisation von FolderSync unter Android



Quellenverzeichnisse

Literaturverzeichnis

AL-Mukhtar, M. M. A./Mardan, A. A. A. (2014): Performance Evaluation of Private Clouds Eucalyptus versus CloudStack, in: International Journal of Advanced Computer Science and Applications, 5. Jg., Nr. 5, S. 108 - 117

Balzert, H. (2011): Lehrbuch der Softwaretechnik, Entwurf, Implementierung, Installation und Betrieb, Lehrbücher der Informatik, 3. Auflage, Heidelberg: Spektrum Akademischer Verlag

Barton, T. (2014): E-Business mit Cloud Computing, Grundlagen | Praktische Anwendungen | verständliche Lösungsansätze, IT-Professional, Wiesbaden: Springer Vieweg

Baun, C. u. a. (2011): Cloud Computing, Web-basierte dynamische IT-Services, Informatik im Fokus, 2. Auflage, Berlin/Heidelberg: Springer-Verlag

Beenken, M. (2013): Der Versicherungsvermittler als Unternehmer, Betriebswirtschaftliche Herausforderungen für Makler und Ausschließlichkeitsvertrieb, 4. Auflage, Karlsruhe: Verlag Versicherungswirtschaft

Boes, A. u. a. (2012): Einführung: Eine global vernetzte Ökonomie braucht die Menschen., Qualifizierung als strategischer Erfolgsfaktor einer nachhaltigen Globalisierung in der IT-Branche, in: Qualifizieren für eine global vernetzte Ökonomie, (Hrsg.: Boes, A. u. a.), Wiesbaden: Springer Gabler, S. 11 - 22

Böhm, M. u. a. (2010): Cloud Compliance, in: Cloud Computing - Was Entscheider wissen müssen, (Hrsg.: Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e. V.), Berlin: o. Verl., S. 88 - 99

Böhnlein, I. (2003): Anwendung von Aspekten der Neuen Institutionenökonomik auf Open Source Software, Produktion, Verfügungsrechte und Transaktionskosten eine theoretische und empirische Untersuchung, o. O.: Diplomica Verlag

Cardy, R. L./Leonard, B. (2011): Performance Management, Concepts, Skills, and Exercises, 2. Auflage, Abingdon: M.E. Sharpe

Diefenbach, S./Brüning, K. T./Rickmann, H. (2013): Effizienz und Effektivität im IT-Outsourcing, KPI-basierte Messung der Strategieumsetzung, in: IT-Outsourcing, (Hrsg.: Rickmann, H./Diefenbach, S./Brüning, K. T.), Berlin/Heidelberg: Springer-Verlag, S. 1 - 24

Finger, P./Zeppenfeld, K. (2009): SOA und WebServices, Informatik im Fokus, Berlin/Heidelberg: Springer-Verlag

Fowler, M. (2003): Patterns für Enterprise Application-Architekturen, Software-Entwicklung, Heidelberg/München/Landsberg/Frechen/Hamburg: mitp-verlag

Golembowska, A. u. a. (2012): Entwicklung eines Modells zur Bewertung von Open Source Produkten hinsichtlich eines produktiven Einsatzes, in: KOS.content, 1, S. 183 - 244

Hansen, M. (2012): Datenschutz im Cloud Computing, in: Daten- und Identitätsschutz in Cloud Computing, E-Government und E-Commerce, (Hrsg.: Borges, G./Schwenk, J.), Berlin/Heidelberg: Springer-Verlag, S. 79 - 96

Kaschuba, A. (2013): Gebrauchstauglichkeit von Informationssystemen, München: GRIN Verlag

Königs, O. (2012): Open Source und Workflow im Unternehmen, Eine Untersuchung von Processmaker, Joget, Bonita Open Solution, uEngine und Activiti, Hamburg: Diplomica Verlag

Küsters, R./Wilke, T. (2011): Moderne Kryptographie, Eine Einführung, Wiesbaden: Vieweg + Teubner

Lopez Lujan, J. M. (2013): An Integral Open Source Software selection model with a case study on IT Infrastructure Monitoring System, o. O.: o. Verl.

MALAYSIAN PUBLIC SECTOR OPEN SOURCE SOFTWARE (OSS) INITIATIVE (o. J.): OPEN SOURCE SOFTWARE (OSS) IMPLEMENTATION GUIDELINES, o. O., S. 63 - 66

Matros, R. (2012): Der Einfluss von Cloud Computing Auf IT-Dienstleister, Eine fallstudienbasierte Untersuchung kritischer Einflussgrößen, Schriftenreihe des Betriebswirtschaftlichen Forschungszentrums/Mittelstand Bayreuth, Wiesbaden: Springer Gabler

Mayerhofer, W. (2009): Einflussfaktoren auf den Abschluss von Versicherungen, und der Beitrag des Versicherungsmaklers, Empirische Marketingforschung, 20. Jg., Wien: facultas.wuv Universitätsverlag

National Institute of Standards and Technology (2001): Announcing the ADVANCED ENCRYPTION STANDARD (AES), Federal Information Processing Standards Publication 197, November, o. O.

Raggi, E./Thomas, K./van Vugt, S. (2011): Beginning Ubuntu Linux, Natty Narwhal Edition, The Expert's Voice in Linux, 6. Auflage, New York: Apress

Saleck, T. (2005a): Chefsache IT-Kosten, Bezahlbare IT, die Leistung sichern, Implementierungshilfen, Edition CIO, 2. Auflage, Wiesbaden: Friedr. Vieweg & Sohn Verlag/GWV Fachverlage

Saleck, T. (2005b): Chefsache Open Source, Kostenvorteile und Unabhängigkeit durch Open Source, Edition CIO, Wiesbaden: Friedr. Vieweg & Sohn Verlag/GWV Fachverlage

Schaaf, A. (2013): Open-Source-Lizenzen, Untersuchung der GPL, LGPL, BSD und Artistic License, Hamburg: Diplomica Verlag

Schill, A./Springer, T. (2012): Verteilte Systeme, Grundlagen und Basistechnologien, eXamen.press, 2. Auflage, Berlin/Heidelberg: Springer Vieweg

Schneider, J. (2012): Datenschutzrechtliche Anforderungen an die Sicherheit der Kommunikation im Internet, in: Daten- und Identitätsschutz in Cloud Computing, E-Government und E-Commerce, (Hrsg.: Borges, G./Schwenk, J.), Berlin/Heidelberg: Springer-Verlag, S. 21 - 42

Seidel, U. (2014): Offener Stapel für Wolkenstürmer, So funktioniert die Cloud-Software Openstack, in: c't, 32. Jg., Nr. 26, S. 14 - 15

The ownCloud developers (2014): ownCloud User Manual, Release 7.0, August, o. O.

Tsolkas, A./Schmidt, K. (2010): Rollen und Berechtigungskonzepte, Ansätze für das Identity- und Access-Management im Unternehmen, Wiesbaden: Vieweg+Teubner

Witt, K.-U. (2014): Algebraische und zahlentheoretische Grundlagen für die Informatik, Gruppen, Ringe, Körper, Primzahltests, Verschlüsselung, Wiesbaden: Springer Vieweg

Yeluri, R./Castro-Leon, E. (2014): Building the Infrastructure for Cloud Security, A Solutions view, New York: Apress

Zehnter, C./Burger, A./Ovtcharova, J. (2012): Key-Performance-Analyse von Methoden des Anforderungsmanagements., KIT scientific reports, Karlsruhe: KIT Scientific Publishing

Verzeichnis der Internet- und Intranetquellen

ActOnMagic Technologies (2014): ActOnCloud SaaS Pricing, http://www.actonmagic.com/pricing.html, Abruf: 29.12.2014

Alexa Internet (2014): How popular is owncloud.org?, http://www.alexa.com/siteinfo/http%3A%2F%2Fowncloud.org, Abruf: 03.12.2014

Alexa Internet (2015a): eucalyptus.com, How popular is eucalyptus.com?, http://www.alexa.com/siteinfo/https%3A%2F%2Fwww.eucalyptus.com, Abruf: 03.12.2014

Alexa Internet (2015b): How popular is openstack.org?, http://www.alexa.com/siteinfo/http%3A%2F%2Fwww.openstack.org, Abruf: 03.01.2015

Apache Software Foundation (o. J.): The Apache Software Foundation, https://issues.apache.org/jira/browse/CLOUDSTACK-5740?jql=project%20%3D%20CLOUDSTACK%20AND%20issuetype%20%3D%20Bug%20AND%20created%20%3E%3D%202014-01-01%20AND%20created%20%3C%3D%202014-12-31, Abruf: 01.01.2015

Apache Software Foundation (2012): Platinum Sponsor(s), http://apache.org/foundation/thanks.html, Abruf: 29.12.2014

Apache Software Foundation (2014a): Apache CloudStack, Open Source Cloud Computing, http://cloudstack.apache.org/, Abruf: 29.12.2014

Apache Software Foundation (2014b): Apache CloudStack Features, http://cloudstack.apache.org/software/features.html, Abruf: 20.01.2015

Apache Software Foundation (2014c): Apache CloudStack: FAQ, http://cloudstack.apache.org/cloudstack-faq.html, Abruf: 03.01.2015

Apache Software Foundation (2014d): Apache CloudStack: Project Membership, http://cloudstack.apache.org/who.html, Abruf: 08.01.2015

Apache Software Foundation (2014e): CloudStack's History, http://cloudstack.apache.org/history.html, Abruf: 30.12.2014

Apache Software Foundation (2014f): Welcome to CloudStack Installation Documentation, http://docs.cloudstack.apache.org/projects/cloudstack-installation/en/latest/, Abruf: 01.01.2015

Appcore (2014): appcore, http://www.appcore.com/, Abruf: 29.12.2014

Bakker, P. (2013): Why use Ephemeral Diffie-Hellman, https://polarssl.org/kb/cryptography/ephemeral-diffie-hellman, Abruf: 04.01.2015

Bartosch, M. (2008): Diffie-Hellman-Schlüsselaustausch, http://www.heise.de/security/artikel/Diffie-Hellman-Verfahren-270980.html, Abruf: 04.01.02014

Böck, H. (2012): Kryptographie nach dem Quantencomputer, http://www.heise.de/tp/artikel/37/37896/1.html, Abruf: 04.01.2014

Bundesamt für Sicherheit in der Informationstechnik (2009): 4 Glossar und Begriffsdefinitionen, https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzKataloge/Inhalt/Glossar/glossar_node.html, Abruf: 04.01.2014

Bundesamt für Sicherheit in der Informationstechnik (2013): Mindeststandard des BSI nach § 8 Abs. 1 Satz 1 BSIG für den Einsatz des SSL/TLS-Protokolls in der Bundesverwaltung, https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Mindeststandards/Mindeststandard_BSI_TLS_1_2_Version_1_0.pdf;jsessionid=B78B6655C8A182C759E21BDD8331933E.2_cid359?__blob=publicationFile, Abruf: 04.01.2015

Burns, C. (2013): Stack wars: OpenStack v. CloudStack v. Eucalyptus, OpenStack has the buzz, CloudStack has the bucks, Eucalyptus has the bonds with Amazon, http://www.networkworld.com/article/2166407/cloud-computing/stack-wars--openstack-v--cloudstack-v--eucalyptus.html, Abruf: 15.01.2015

Büst, R. (2012): Plattformen für die Private Cloud, http://www.computerwoche.de/a/plattformen-fuer-die-private-cloud,2514114,2, Abruf: 20.01.2015

Canonical (2014a): Cinder, https://bugs.launchpad.net/cinder/+bugs?field.searchtext=&search=Search&field.status%3Alist=NEW&field.status%3Alist=WONTFIX&field.status%3Alist=EXPIRED&field.stat, Abruf: 19.01.2015

Canonical (2014b): OpenStack Compute (Nova), https://bugs.launchpad.net/nova/+bugs?field.searchtext=&search=Search&field.status%3Alist=NEW&field.status%3Alist=WONTFIX&field.status%3Alist=EXPIRED&field.statu, Abruf: 19.01.2015

Canonical (2014c): OpenStack Dashboard (Horizon), https://bugs.launchpad.net/horizon/+bugs?field.searchtext=&search=Search&field.status%3Alist=NEW&field.status%3Alist=WONTFIX&field.status%3Alist=EXPIRED&field.st, Abruf: 19.01.2015

Canonical (2014d): OpenStack Neutron (virtual network service), https://bugs.launchpad.net/neutron/+bugs?field.searchtext=&search=Search&field.status%3

Alist=NEW&field.status%3Alist=WONTFIX&field.status%3Alist=EXPIRED&field.st, Abruf: 19.01.2015

Canonical (2014e): OpenStack Object Storage (Swift), https://bugs.launchpad.net/swift/+bugs?field.searchtext=&search=Search&field.status%3Alist=NEW&field.status%3Alist=WONTFIX&field.status%3Alist=EXPIRED&field.statu, Abruf: 19.01.2015

Eikenberg, R. (2014): Kaspersky-Schutzsoftware senkt Sicherheit von SSL-Verbindungen, http://www.heise.de/security/meldung/Kaspersky-Schutzsoftware-senkt-Sicherheit-von-SSL-Verbindungen-2482344.html, Abruf: 13.12.2014

Eucalyptus Systems (o. J.): Eucalyptus, https://eucalyptus.atlassian.net/browse/EUCA-10276?jql=project%20%3D%20EUCA%20AND%20issuetype%20%3D%20Bug%20AND%20created%20%3E%3D%202014-01-01%20AND%20created%20%3C%3D%202014-12-31, Abruf: 01.01.2015

Eucalyptus Systems (2014a): Eucalyptus Datasheet, https://www.eucalyptus.com/sites/all/files/ds-eucalyptus-iaas.en.pdf, Abruf: 17.01.2015

Eucalyptus Systems (2014b): Eucalyptus Documentation, Official Documentation for Eucalyptus Cloud, https://www.eucalyptus.com/docs/eucalyptus/4.0.2/index.html#shared/index.html, Abruf: 01.01.2015

Eucalyptus Systems (2014c): Eucalyptus Systems Introduces Release 4.0 Enabling Seamless Growth Of Large Scale Private And Hybrid Clouds, Leading Open Source AWS-Compatible Cloud Provider Releases Eucalyptus 4.0 With Advanced Networking, Storage And Security Capabilities For Enterprise Clouds, https://www.eucalyptus.com/news/eucalyptus-systems-introduces-release-40-enabling-seamless-growth-large-scale-private-and, Abruf: 17.01.2015

Eucalyptus Systems (2015a): Consulting / System Integration Partners, https://www.eucalyptus.com/partners/consulting-si, Abruf: 02.01.2015

Eucalyptus Systems (2015b): Contact Us, We Want To Hear From You!, https://www.eucalyptus.com/about/contact?OfferDetails=Contact%20Us%20Form&OfferURL=http%3A//go.eucalyptus.com/Contact-Us.html, Abruf: 08.01.2015

Eucalyptus Systems (2015c): Eucalyptus, https://www.eucalyptus.com/, Abruf: 03.01.2015

Eucalyptus Systems (2015d): Eucalyptus Cloud Features, https://www.eucalyptus.com/eucalyptus-cloud/iaas/features, Abruf: 06.01.2015

Eucalyptus Systems (2015e): Platform Partners, https://www.eucalyptus.com/partners/platform, Abruf: 29.12.2014

Eucalyptus Systems (2015f): Reseller / OEM Partners, https://www.eucalyptus.com/partners/reseller-oem, Abruf: 02.01.2015

Eucalyptus Systems (2015g): Technology Partners, https://www.eucalyptus.com/partners/technology, Abruf: 29.12.2014

Eymann, T. (2014): Cloud Computing, http://www.enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/lexikon/uebergreifendes/Kontext-und-Grundlagen/Markt/Softwaremarkt/Geschaftsmodell-%28fur-Software-und-Services%29/Cloud-Computing/index.html, Abruf: 21.11.2014

Gallagher, I./Condron, S./Watkins, S. (2014): Exposed: Barclays account details for sale as 'gold mine' of up to 27,000 files is leaked in worst breach of bank data EVER, http://www.dailymail.co.uk/news/article-2554875/Barclays-account-details-sale-gold-27-000-files-leaked.html, Abruf: 01.01.2015

GitHub (2015a): apache / cloudstack, Releases, https://github.com/apache/cloudstack/releases, Abruf: 30.12.2014

GitHub (2015b): apache/cloudstack, https://github.com/apache/cloudstack, Abruf: 16.01.2015

GitHub (2015c): eucalyptus/eucalyptus, https://github.com/eucalyptus/eucalyptus, Abruf: 16.01.2015

GitHub (2015d): eucalyptus/eucalyptus, Releases, https://github.com/eucalyptus/eucalyptus/releases, Abruf: 06.01.2015

GitHub (2015e): eucalyptus/eucalyptus-database-server, https://github.com/eucalyptus/eucalyptus-database-server, Abruf: 17.01.2015

GitHub (2015f): openstack/openstack, https://github.com/openstack/openstack, Abruf: 16.01.2015

GitHub (2015g): owncloud / core, Issues, https://github.com/owncloud/core/issues?q=label%3Abug+created%3A2014-01-01.2014-12-31, Abruf: 01.01.2015

GitHub (2015h): owncloud/core, https://github.com/owncloud/core, Abruf: 16.01.2015

Gude, S. (2014): Zwei-Faktor-Authentifizierung (2FA) für ownCloud 6 - Update: Kompatibilität zu ownCloud 7 (09.08.2014), http://www.cy-man.de/?x=entry:entry140330-121225, Abruf: 06.01.2015

Hill, Z. (2013): Eucalyptus Storage, https://github.com/eucalyptus/eucalyptus/wiki/Storage, Abruf: 17.01.2015

Host Europe (o. J.): Wir machen Ihr Business agil, Mit Ihrer offenen OpenStack-Plattform, https://www.hosteurope.de/Cloud/OpenStack/?gclid=CjwKEAiAt4mlBRDXwt_m9ICU4DcSJAAS_X0WGJS2vNQKAwi96nZgwBlC7RXEFo08d-7EJVpOj9Wh4BoC4xDw_wcB, Abruf: 29.12.2014

Huang, A./Childers, C./Kinsella, J. (2013): Releases, https://cwiki.apache.org/confluence/display/CLOUDSTACK/Releases, Abruf: 13.01.2015

Huger, J. W. (2014): Top 10 open source projects of 2014, http://opensource.com/business/14/12/top-10-open-source-projects-2014, Abruf: 15.01.2015

ITwissen (o. J.): Community Cloud, http://www.itwissen.info/definition/lexikon/Community-Cloud-community-cloud.html, Abruf: 10.12.2014

Kangaroot (2015): Get your own secure online storage with ownCloud, http://kangaroot.net/secure-online-storage-owncloud, Abruf: 02.01.2015

Kannan, H. (2013): S3-based secondary Storage, https://cwiki.apache.org/confluence/display/CLOUDSTACK/S3-based+secondary+Storage, Abruf: 17.01.2015

Karlitschek, F. (2010): ownCloud 1.0 is here, http://karlitschek.de/2010/06/owncloud-1-0-is-here/, Abruf: 03.01.2015

Maguire, J. (2013): What is Private Cloud?, http://www.datamation.com/cloud-computing/what-is-private-cloud.html, Abruf: 21.11.2014

National Institute of Standards and Technology (o. J.): Descriptions of SHA-256, SHA-384, and SHA-512, http://csrc.nist.gov/groups/STM/cavp/documents/shs/sha256-384-512.pdf, Abruf: 04.01.2014

Network Working Group (2008): The Transport Layer Security (TLS) Protocol Version 1.2, https://tools.ietf.org/html/rfc5246, Abruf: 04.01.2015

o. V. (o. J.a): Changelog ownCloud, https://owncloud.org/releases/Changelog, Abruf: 06.01.2015

o. V. (o. J.b): Diffie-Hellman-Merkle-Schlüsselaustausch, http://www.elektronik-kompendium.de/sites/net/1909031.htm, Abruf: 04.01.2015

o. V. (2013): authentifizieren, http://www.duden.de/rechtschreibung/authentifizieren, Abruf: 04.01.2015

op5 (o. J.): Eucalyptus, http://www.op5.com/explore-op5-monitor/features/cloud-monitoring/eucalyptus/, Abruf: 20.01.2015

OpenStack (o. J.a): Chapter 18. Identity, http://docs.openstack.org/security-guide/content/ch024_authentication.html, Abruf: 06.01.2015

OpenStack (o. J.b): Companies Supporting The OpenStack Foundation, Platinum Members, http://www.openstack.org/foundation/companies/, Abruf: 29.12.2014

OpenStack (o. J.c): Documentation (Current), http://docs.openstack.org/, Abruf: 29.12.2014

OpenStack (o. J.d): German, http://docs.openstack.org/de/, Abruf: 29.12.2014

OpenStack (o. J.e): Neutron/quantum-network-connection-monitoring-service, https://wiki.openstack.org/wiki/Neutron/quantum-network-connection-monitoring-service, Abruf: 20.01.2015

OpenStack (o. J.f): OpenStack, http://www.openstack.org/community/, Abruf: 08.01.2015

OpenStack (o. J.g): OpenStack, https://ask.openstack.org/en/questions/, Abruf: 03.01.2015

OpenStack (o. J.h): OpenStack Community Q&A, http://www.openstack.org/projects/openstack-faq/, Abruf: 03.01.2015

OpenStack (o. J.i): OpenStack Networking, PLUGGABLE, SCALABLE, API-DRIVEN NETWORK AND IP MANAGEMENT, http://www.openstack.org/software/openstack-networking/, Abruf: 17.01.2015

OpenStack (o. J.j): OpenStack Storage, OBJECT AND BLOCK STORAGE FOR USE WITH SERVERS AND APPLICATIONS, http://www.openstack.org/software/openstack-storage/, Abruf: 17.01.2015

OpenStack (o. J.k): Releases, https://wiki.openstack.org/wiki/Releases, Abruf: 29.12.2014

OpenStack (o. J.l): Releases, https://wiki.openstack.org/wiki/Releases, Abruf: 06.01.2015

ownCloud (o. J.a): Accessing your Files (Web Interface), http://doc.owncloud.org/server/6.0/user_manual/files/filesweb.html, Abruf: 20.01.2015

ownCloud (o. J.b): Documentation Overview, http://doc.owncloud.org/, Abruf: 03.01.2015

ownCloud (o. J.c): Manual Installation, http://doc.owncloud.org/server/6.0/admin_manual/installation/installation_source.html, Abruf: 08.12.2014

ownCloud (o. J.d): ownCloud 7.0 User Documentation, Welcome to ownCloud: your self-hosted file sync and share solution, http://doc.owncloud.org/server/7.0/user_manual/, Abruf: 17.01.2015

ownCloud (o. J.e): ownCloud Forums, A board to discuss all kind of ownCloud topics, http://forum.owncloud.org/, Abruf: 03.01.2015

ownCloud (o. J.f): ownCloud: File Sharing Under Your Control, https://owncloud.com/community-enterprise/, Abruf: 08.12.2014

ownCloud (2014a): Downloads, https://owncloud.com/download/, Abruf: 23.12.2014

ownCloud (2014b): ownCloud Enterprise Edition Features, https://owncloud.com/features/#Antivirus, Abruf: 26.12.2014

ownCloud (2015a): Contact, Active contributors on the ownCloud blog roll:, http://owncloud.org/contact/, Abruf: 08.01.2015

ownCloud (2015b): Frequently Asked Questions, https://owncloud.org/faq/, Abruf: 03.01.2015

ownCloud (2015c): ownCloud, https://owncloud.org/install/, Abruf: 06.01.2015

ownCloud (2015d): ownCloud Partner, https://owncloud.com/de/partner/, Abruf: 08.01.2015

ownCloud (2015e): ownCloud Server Changelog, http://owncloud.org/changelog/, Abruf: 30.12.2014

Paysant, P. (2014): Dashboard 1.2, ownCloud Tool, https://apps.owncloud.com/content/show.php/Dashboard?content=167139&PHPSESSID=32c088e9628c43d07e515bedfe7dd70d, Abruf: 20.01.2015

Piekenbrock, D./Fehling, C./Leymann, F. (o. J.): Elastizität, http://wirtschaftslexikon.gabler.de/Archiv/55255/elastizitaet-v9.html, Abruf: 08.01.2015

Porteck, S. (2013): Alles auf Abruf, Kalender und Kontakte über OwnCloud synchronisieren, http://www.heise.de/ct/artikel/Alles-auf-Abruf-1908923.html, Abruf: 06.01.2015

Rackspace (o. J.): Rackspace Managed Cloud, Public, Private und Hybrid Clouds powered by OpenStack, http://www.rackspace.com/de/cloud?gclid=CjwKEAiAt4mlBRDXwt_m9ICU4DcSJAAS_X0WlkmTDrzDzGUpJwKv6zKmZcf_tGDft1cKBYkFM_bGLRoCFTLw_wcB&utm_medium=ppc&utm_source=googlede&utm_campaign=2_pro_gen_cloud&utm_term=openstack&utm_content=gen&s_kwcid=AL!2805!3!49713588877!p!!g!!openstack&ef_id=Uvvb9wAABXsZclCw:20141230173529:s, Abruf: 29.12.2014

RES Software (o. J.): Consumerization of IT and the ITaaS Mindset, http://www.ressoftware.com/itaas, Abruf: 28.12.2014

RP Online (2010): Deutsche geben mehr Geld für Versicherungen aus, Trotz Wirtschaftskrise, http://www.rp-online.de/wirtschaft/unternehmen/deutsche-geben-mehr-geld-fuer-versicherungen-aus-aid-1.2323324, Abruf: 01.01.2015

Schießle, B. (2013): Introduction to the new ownCloud Encryption App, http://blog.schiessle.org/2013/05/28/introduction-to-the-new-owncloud-encryption-app/, Abruf: 01.12.2014

Schmidt, J. (2013): Zukunftssicher verschlüsseln mit Perfect Forward Secrecy, http://www.heise.de/security/artikel/Zukunftssicher-Verschluesseln-mit-Perfect-Forward-Secrecy-1923800.html, Abruf: 04.01.2014

Shepherd, D. (2013): Plug-ins, Modules, and Extensions, https://cwiki.apache.org/confluence/display/CLOUDSTACK/Plug-ins%2C+Modules%2C+and+Extensions, Abruf: 17.01.2015

Sokolov, D. A. J. (2014): Spearphishing: Jeder Fünfte geht in die Falle, http://heise.de/-2461982, Abruf: 01.12.2014

Steinmetz, D. u. a. (2012): Cloud Computing Performance Benchmarking and Virtual Machine Launch Time, http://sigite2012.sigite.org/wp-content/uploads/2012/09/p89.pdf, Abruf: 15.01.2015

The Register (2010): NASA and Rackspace open source cloud fluffer, OpenStack targets one million machine Nebula, http://www.theregister.co.uk/2010/07/19/nasa_rackspace_openstack/, Abruf: 30.12.2014

Thome, N./Haynberg, R. (2014): Perfect Forward Secrecy - was ist das?, http://www.computerwoche.de/a/perfect-forward-secrecy-was-ist-das,3063860, Abruf: 04.01.2015

Vollmer, A. (2014): Umfrage: Deutschen verunsichert wegen NSA-Überwachung, http://www.datenschutz.de/news/detail/?nid=6709, Abruf: 21.11.2014

Wolters, F. u. a. (2009): Cloud Computing und Open Source, http://winfwiki.wi-fom.de/index.php/Cloud_Computing_und_Open_Source#Eucalyptus, Abruf: 06.01.2015

Zwior, T. (2014): Vertrauen in Versicherungsbranche gering, http://www.versicherungsbote.de/id/4806081/Versicherung-Versicherungsmakler-Kundenvertrauensindex-Vertrauen, Abruf: 29.12.2014

Gesprächsverzeichnis

Anonymisiert, U. (2014): Bereichsleiter Anwendungsentwicklung, _Versicherung, Stuttgart, persönliches Gespräch am 08.10.2014 in Stuttgart

Ihlein, J. (2014): Diplom-Mathematiker und Maklerbetreuer, Wüstenrot und Württembergische AG, Stuttgart, persönliches Gespräch am 27.12.2014 in Leonberg

Kionga, C. (2014): Principal Consultant, Bereichsleitung IT-Security Competence Center, Bechtle Internet Security & Services, Bechtle GmbH & Co. KG, Neckarsulm, E-Mail-Korrespondenz am 21.11.2014



Open Source Lösungen zur Spracherkennung

Schriftliche Ausarbeitung

im Rahmen der Lehrveranstaltung „Integrationsseminar“

für das Kompetenzzentrum Open Source (KOS)

Vorgelegt von

Tobias Jacoby, Silvestro Perpetuo

Katharina Junge, Franziska Troszt

am 23.01.2015

Fakultät Wirtschaft

Studiengang Wirtschaftsinformatik

WWI2012V

Inhaltsverzeichnis

Abkürzungsverzeichnis

ASR	**A**achen University **S**peech **R**ecognition System
CMU	**C**arnegie **M**ellon **U**niversity
HMM	**H**idden **M**arkov **M**odell
HTK	**H**idden Markov **T**ool**k**it
MLLR	**m**aximum **l**ikelihood **l**inear **r**egression
RWTH	**R**heinisch-**W**estfälische **T**echnische **H**ochschule
WA	**W**ort**a**kkuratheit
WK	**W**ort**k**orrektheit

Tabellenverzeichnis

Abbildungsverzeichnis

1 Einleitung

Seit 2012 hat sich die Anzahl an Smartphone-Nutzern fast verdoppelt. Eine Studie von Februar 2014 zeigt, dass 40,4 Millionen Deutsche ein Smartphone besitzen. Das sind 50%[1] der gesamten deutschen Bevölkerung.[2] Vor allem hochwertige Smartphones, wie das aktuelle Apple iPhone 6 oder Samsung Galaxy S5, besitzen eine Sprachsteuerung. Somit hat ungefähr die Hälfte der deutschen Bevölkerung Zugriff auf Spracherkennungsdienste.

Der Einsatz von Spracherkennungssystemen bei Smartphones spiegelt wider, welche großen Fortschritte in der Spracherkennung gemacht wurden und wie sich diese Systeme zum gleichberechtigten Eingabeinstrument neben Tastatur und Maus entwickeln. Der Beginn der Spracherkennung war im Jahre 1984 als IBM das erste Spracherkennungssystem vorstellte. Dieses System konnte mithilfe eines Großrechners ca. 5.000 englische Einzelworte erkennen. Im selben Jahr wurde ein fortschrittlicheres System für tragbare PCs von der Firma Dragon Systems entwickelt, das im Jahr 1990 als Dragon Dictate System in den USA vorgestellt wurde. 1986 erschien das IBM-System TANGORA 4 zunächst in englischer Sprache, das im Jahre 1991 auf der CeBIT auch in deutscher Sprache vorgestellt wurde. TANGORA 4 umfasste 20.000 bis 30.000 Worte. Das erste auf einem PC anwendbare Produkt war die IBM Speech Server Series. Dies war eine Client-Server-Lösung für das Betriebssystem AIX, das besonders in Krankenhäusern eingesetzt wurde. Ein Durchbruch für den breiten Massenmarkt gelang IBM im Jahre 1993 mit dem Namen IBM Personal Dictation System, das später in IBM VoiceType-Diktiersystem umbenannt wurde. Dieses System kostete nur 1.000 Euro. Zuvor lagen die Kosten für Spracherkennungssysteme bei ca. 10.000 Euro. 1995 wurde eine deutsche Version namens VoiceType-Diktiersystem mit speziellem Fachvokabular für Anwälte und Mediziner vorgestellt.[3]

Das Platzen der Internet-Blase im Jahre 2000 führte zu einer Unterbrechung der Weiterentwicklung von Spracherkennungssystemen.[4] Neue Bewegungen in der Spracherkennungsentwicklung brachte das System Siri von Apple. Anfang des Jahres 2010 wurde Siri als iPhone-App veröffentlicht.[5] Der letztendliche Durchbruch von dem System Siri war Ende des Jahres 2011, als das iPhone 4S mit dem integrierten Spracherkennungssystem Siri vorgestellt wurde. Daraufhin folgten weitere Entwicklungen in der Spracherkennungsbranche, wie zum Beispiel in 2012 das Spracherkennungssystem Google-Now für Smartphones.[6]

[1] In Deutschland leben 80,8 Millionen Menschen, somit sind 40,4 Millionen deutsche Smartphone-Nutzer 50% der Bevölkerung.
[2] Vgl. WeltN24 GmbH (2014); vgl. Statista (2015)
[3] Vgl. Linguatec (2014)
[4] Vgl. Dernbach, C. (2012)
[5] Vgl. CHIP Digital GmbH (2013)
[6] Vgl. Roodsari, A. V. (2014)

Die Problemstellung dieser Arbeit bezieht sich auf eine Applikation für den Vertrieb der _Versicherung-Versicherungen. Mit Hilfe der Applikation können die Vertreter unter anderem Versicherungsdaten ihrer Kunden abrufen und Schadenfälle aufnehmen. Bezüglich der Schadenfälle werden Schadensberichte schriftlich festgehalten. Um den Prozess der Schadensaufnahme zu erleichtern und die Effizienz zu steigern, soll es möglich sein, die Schadensberichte mithilfe eines Spracherkennungsprogramms aufzunehmen. Spracherkennung definiert sich als „automatisches bzw. maschinelles Erkennen gesprochener Sprache“[7]. Spracherkennungsprogramme sind „Programm[e], [die] [..] die Spracherkennung eines Computers ermöglich[en]“[8]. Die Spracherkennungsprogramme sollen die gesprochenen Berichte aufnehmen und in Text umwandeln. Der Text soll schließlich in der Applikation gespeichert werden. Das Spracherkennungsprogramm soll folglich in die vorhandene Applikation der _Versicherung integriert werden.

Die Anforderungen des Projekts beziehen sich hauptsächlich auf die Identifizierung und Bewertung von Open Source-Spracherkennungstools, um dieses Problem zu lösen. Neben allgemeinen Informationen über Spracherkennungstools wird analysiert, wie diese Tools technisch funktionieren. Anschließend werden verfügbare Open Source-Tools identifiziert und kurz beschrieben. Anhand eines selbst erstellten Kriterienkataloges werden die Open Source-Tools schließlich analysiert und bewertet. Zusätzlich wird der Kriterienkatalog auf das kommerziell verfügbare Spracherkennungstool Dragon NaturallySpeaking angewandt, um es ebenfalls zu bewerten und mit den ausgewählten Open Source-Tools zu vergleichen. Es soll ein Open Source-Tool identifiziert werden, das sich für die Aufnahme von Schadensberichten eignet. Zuletzt wird eines der behandelten Spracherkennungstools getestet.

[7] Bibliographisches Institut GmbH (2013b)
[8] Bibliographisches Institut GmbH (2013c)

2 Spracherkennung

Wie bereits erwähnt wurde, ist die Umwandlung menschlicher Sprache in ein für Computer verständliches Format seit mehreren Jahrzehnten ein begehrtes Ziel. Dieses Ziel wird allgemein Spracherkennung genannt. Hierbei wird außer Acht gelassen, dass Spracherkennung abhängig von der Aufgabenstellung verschiedene Ziele und somit verschiedene Teilgebiete beinhaltet. Deshalb ist es ratsam, zu Beginn eines Projektes klar zu definieren, welche Teilgebiete relevant sind und welche Teilgebiete zunächst außer Acht gelassen werden, um sich in der Fülle an Informationen zu Recht zu finden. Im Folgenden sollen zunächst die Teilgebiete aufgezeigt werden. Anschließend folgt eine Erläuterung möglicher Schwierigkeiten, welchen man in einem Spracherkennungssystem begegnen kann.

2.1 Teilgebiete der Spracherkennung

Die **Spracherkennung** an sich bildet ein eigenes Teilgebiet der kompletten Materie. Sie beschäftigt sich mit der Aufgabe zu verarbeiten, was gesagt wurde.[9] Dieses Teilgebiet wird anhand des Anwendungsfalls in drei Kategorien aufgeteilt. Hier finden sich zum einen die *Diktiersysteme* wieder, welche für Anwender, die häufig in flüssigem Stil diktieren, einen großen Nutzer erbringen. Eine weitere Kategorie bildet die *Gerätesteuerung*, welche dazu dient eine zusätzliche Eingabemöglichkeit zur Verfügung zu stellen. Angefangen bei alltäglichen Dingen wie die Namenswahl in einem Telefonbuch, schafft die Gerätesteuerung zum Beispiel als aktive Kommunikationshilfe für Menschen mit Behinderungen oder in der Medizin bei der Steuerung eines Operationsmikroskops einen besonders hohen Nutzen. Eingesetzt in Infotainmentkomponenten wie Navigationssystemen, Mobiltelefonen und Autoradios ermöglicht die Gerätesteuerung zudem eine Erhöhung der Verkehrssicherheit. Beispielsweise muss der Anwender während des Fahrens eines KFZs seine Sicht nicht auf das zu bedienende Gerät richten. Die dritte Kategorie bildet das sogenannte *Voice Portal*, welchem die Menschen des Öfteren im Alltag begegnen. Hierbei handelt es sich um Systeme, die dem Anrufer durch Sprachdialog verschiedene Dienste zur Verfügung stellen. In Call-Centern werden diese zum Beispiel dazu eingesetzt, durch eine Vorfilterung des Kundenanliegens den richtigen Ansprechpartner zu finden oder dem Kunden direkt eine automatische Antwort zu geben. Weitere Einsatzgebiete finden sich in Auskunftssystemen, Bestell- und Reservierungssystemen, Telefon-Banking und telefonischen Gewinnspielen.[10]

Das sogenannte **Keyword-Spotting** wird oft eingesetzt, um Audio- und Filmarchive auf ein bestimmtes Thema zu durchsuchen und um Telefongespräche abzuhören. Hierbei wird das

[9] Vgl. Euler, S. (2006), S.15
[10] Vgl. ebenda, S.17

Audiosignal auf ein bestimmtes Stichwort durchsucht. Betrachtet man große Medienkonzerne, welche stets einen passenden Beitrag aus dem Archiv zu den aktuellen Themen abspielen, bietet das Keyword-Spotting ein gutes, ergänzendes Suchmittel.[11]

Die **Sprecheridentifikation bzw. Sprechererkennung** befasst sich mit der Aufgabe einen Sprecher aus einer bestimmten Gruppe von Leuten zu identifizieren. Hierzu wird das individuelle Sprachmuster des Sprechers analysiert. Anwendung findet solch ein System zum Beispiel in Meetings, in welchen das Gesprochene zur nachträglichen Betrachtung aufgezeichnet wird und eventuell darauf durch ein Spracherkennungssystem in ein Textdokument umgewandelt wird.[12]

Etwas genauer nimmt es hier die **Sprecherverifikation**. Diese befasst sich mit der Aufgabe festzustellen, ob es sich beim Sprecher tatsächlich um eine bestimmte Person X handelt. Das Ergebnis der Analyse endet folglich bei einem positiven oder negativen Ergebnis. Hierbei gibt es zwei Vorgehensweisen: Zum einen das *textabhängige Verfahren* und zum anderen das *textunabhängige Verfahren*. Diese unterscheiden sich darin, dass dem Sprecher entweder ein zu sprechender Text vorgegeben wird oder nicht. Anwendung findet die Sprecherverifikation zum Beispiel in Zugangskontrollen als zusätzliche Sicherheitsstufe. Hier sollte man die Sprecherverifikation tatsächlich nur als zusätzliche Sicherheitsstufe betrachten, da ein offensichtlicher Schwachpunkt die Aufnahme der Äußerung des Sprechers darstellt. Diese könnte, wenn die Aufnahme qualitativ sehr hochwertig ist, dazu genutzt werden, um eine Sicherheitsstufe ordnungswidrig zu umgehen.[13]

Ähnlich ist das sogenannte **Speaker-Indexing**. Hier geht es zwar auch um einen bestimmten Sprecher, jedoch ist die Aufgabe eine andere. Zum Beispiel wird in Audioarchiven nach bestimmten Sprechern gesucht. Somit bildet das Speaker-Indexing eine Mischung aus der Sprecheridentifikation, wo es darum geht einen Sprecher aus einer Gruppe zu identifizieren, und der Sprecherverifikation, wo festgestellt wird, ob es sich tatsächlich um den gesuchten Sprecher handelt.[14]

Die **Sprachidentifikation** befasst sich weniger mit der Person sondern mehr mit einer bestimmten Eigenschaft des Sprechers. Anhand eines zu verarbeitenden Audiosignals wird analysiert, in welcher Sprache das Signal vorliegt. Einen großen Nutzen verleihen solche Systeme zum Beispiel bei der Durchsuchung eines großen, multilingualen Audio- und Filmarchivs nach Beiträgen in einer bestimmten Sprache.[15]

[11] Vgl. Euler, S. (2006), S. 16
[12] Vgl. ebenda, S. 16
[13] Vgl. ebenda, S. 16
[14] Vgl. ebenda, S. 16
[15] Vgl. ebenda, S. 16

Alle der oben genannten Kategorien stehen keinesfalls für sich allein da. Im Gegenteil, man findet in großen Spracherkennungssystemen oft Kombinationen der verschiedenen Teilgebiete. Jedoch gibt es auch Anwendungen, die für eine bestimmte Aufgabenstellung spezialisiert sind, wie zum Beispiel bei Diktiersystemen. Das Gegenstück der Spracherkennung ist **die Sprachsynthese**. Sie befasst sich mit der Aufgabe, digitalen Text in Audiosignale umzuwandeln. Hierbei wird auf aufgezeichnete Äußerungen professioneller Sprecher zurückgegriffen und spielt diese bei Bedarf ab. Problematisch ist die Beschränkung des Wortschatzes. Nicht für alle Wörter kann auf die zuvor erstellte Aufzeichnung zurückgegriffen werden.[16]

2.2 Schwierigkeiten der Spracherkennung

Sprachen leben vom ständigen Wandel. Ihr Entwicklungsprozess hört nie auf. Es gibt stets neue Wortschöpfungen, welche sich oft durch Entlehnungen aus anderen Sprachen ergeben. Hinzu kommen meist nicht wenige Grundregeln, welche etliche Sonderfälle und Ausnahmen enthalten.[17] Mit solchen Schwierigkeiten kämpfen die Experten der Spracherkennung bereits seit Jahrzehnten. Im Allgemeinen werden die Schwierigkeiten in sprachliche und technische Bereiche gegliedert.

2.2.1 Sprachliche Schwierigkeiten

Neben dem stetigen Wandel der Sprache und den vielen Grundregeln stellt ein für die menschliche Intelligenz alltägliches Phänomen ein Hindernis für die Spracherkennung dar. Gemeint ist, dass der Sinn eines Satzes meist nur durch die Betonung, den Kontext oder durch Hintergrundwissen verständlich ist. Ein Beispiel hierfür ist das Sprichwort

„Wenn hinter Fliegen Fliegen fliegen, dann fliegen Fliegen Fliegen nach".

Selbst wir Menschen müssen gegebenenfalls ein zweites Mal hinhören, wenn wir diesen Satz das erste Mal gesagt bekommen. Der Sinn des Satzes wird aufgrund der Betonung des Satzes, dem Kontext der Wörter und dem Hintergrundwissen über die Rechtschreibungs- und Grammatikregeln verstanden. Es ist ersichtlich, dass in dem Satz nicht ständig dasselbe Wort genutzt wird.[18] Diese Probleme lassen sich in die Problemfelder Ambiguität, Variabilität und Komplexität der Sprache einteilen.

Die **Ambiguität** beschreibt das Problem der Mehrdeutigkeiten von gleichklingenden Wörtern oder Sätzen. Durch die Homophonie vieler Wörter sind diese meist nur durch die oben genannten drei Kriterien unterscheidbar. Ein Beispiel hierfür sind die beiden Wörter „Rad" und

[16] Vgl. Euler, S. (2006), S. 17
[17] Vgl. ebenda, S. 2
[18] Vgl. ebenda, S. 3

„Rat". Ohne Kontext ist nicht ersichtlich, ob eine gut gemeinte Empfehlung, ein alltägliches Fortbewegungsmittel oder vielleicht sogar ein kreisrunder Gegenstand gemeint ist. Diese Problematik tritt im Alltag oft auf, da Menschen versuchen eine Nachricht mit dem Minimum des erforderlichen Aufwandes zu vermitteln. Für ein Spracherkennungssystem kann dies ein in Eigenregie unüberwindbares Hindernis sein. Deshalb sollte dem Anwender eines Spracherkennungssystem klar gemacht werden, dass eine sehr hohe Sprachqualität für die Nutzung erforderlich ist.

Die **Variabilität** der Sprache beschäftigt sich mit der Tatsache, dass jedes Wort von jedem Menschen individuell ausgesprochen wird. Somit ist ein sogenannter Ausspracheprototyp, an welchem sich ein Spracherkennungssystem orientieren kann, schwer ermittelbar. Diese unterschiedlichen Aussprachegründe haben zum einen Ursachen an den Merkmalen des Sprechers selbst, als auch an situationsbedingten Ursachen.

Eine Zusammenstellung hierzu ist in Tabelle 1 zu finden.[19]

Ursachen	Merkmale
Sprechermerkmale	Dividuell Alter, Geschlecht, Gesundheitszustand, Vokaltraktanatomie (Lautbildung),… Habituell Dialekt, Soziolekt (Gruppensprache),…
Sprechweise	Tempo, Artikulation, Emotionen,…
Aufnahmekanal	Mikrophonqualität, Entfernung zum Mikrophon, akustische Störquellen,…
Kontextuelle Aussprachevariation	-

Tabelle 1: Ursachen der Sprachvariabilität[20]

2.2.2 Technische Schwierigkeiten

Auch technisch gesehen ergeben sich diverse Schwierigkeiten für die Umsetzung eines Spracherkennungssystems. Das häufigste Problem ist die akustische Umgebung bei der Aufnahme eines Audiosignals, welches durch ein Spracherkennungssystem verarbeitet wer-

[19] Vgl. Schukat-Talamazzini, E.G. (1995), S.8 ff.
[20] Vgl. ebenda, S.8 ff.

den soll. Störende Hintergrundgeräusche verzerren das Signal, wodurch dieses schlechter verarbeitet werden kann.[21]

Doch auch wenn diese Hürde bewältigt wurde, muss ein Spracherkennungssystem so programmiert sein, dass es mit der **Kontinuität** des Sprachschalls zurechtkommt. Im Sprachschall gibt es keine eindeutig erkennbaren Grenzen zwischen Lauten, Silben oder Wörtern. Lediglich zwischen Sätzen lassen sich manchmal durch intonationsbedingte Stillintervalle Phrasengrenzen erkennen.

Wenn eine klare Trennung der Wörter möglich ist, stellt die **Komplexität** der Sprache ein weiteres Problem dar. Die hohe Datenrate der Signalabtastwerte eines Audiosignals und die ausartende Kombinatorik der Wortfolgen, durch welche sich Spracherkennungssysteme orientieren, um das auf ein anderes Wort folgende Wort zu bestimmen, verlangen eine hohe Rechenleistung.

Zudem kommt noch, dass die Ausspracheprototypen einen hohen Speicherbedarf haben.[22] Für einen Server im Rechenzentrum eines Unternehmens mag dies zunächst kein großes Problem darstellen. Werden die heutzutage üblichen Mobiltelefone betrachtet, kann es zu Problemen mit dem auf wenige Gigabyte beschränkten Speicherplatz geben.

2.2.3 Unterscheidungskriterien für Herausforderungen der Spracherkennung

Die vielen unterschiedlichen Anwendungsfälle der Spracherkennung verlangen verschiedene Anforderungen an ein Spracherkennungssystem. Diese Anforderungen sind meist an diverse Schwierigkeiten gebunden. Neben den zuvor erwähnten Kriterien, zeigt Tabelle 2 die drei üblichen Unterscheidungskriterien für Spracherkennung.

	Einfach	**Schwer**
Sprechweise	einzelne Wörter	Sätze
Wortschatz	kleiner Wortschatz	großer Wortschatz
Benutzerkreis	sprecherabhängig	sprecherunabhängig

Tabelle 2: Unterscheidungskriterien für Spracherkennung[23]

Vom **Wortschatz** hängt die Erkennungssicherheit des Spracherkennungssystems ab. Ab einer Anzahl von 1000 Wörtern wird von einem großen Wortschatz gesprochen. Ein großer

[21] Vgl. Euler, S. (2006), S.18
[22] Vgl. Schukat-Talamazzini, E.G. (1995), S.10 ff.
[23] Enthalten in: Euler, S. (2006), S.18

Wortschatz bedeutet jedoch, dass sich in der Bestimmung der Wortfolge Schwierigkeiten ergeben können. Viele Diktiersysteme helfen sich beispielsweise dadurch, dass aus der Menge aller möglichen Wörter eine Teilmenge bestimmt wird. Die Teilmenge enthält alle möglichen Wörter, die auf das vorangegangene Wort folgen könnten. Diese Vorgehensweise aber verschafft in Anwendungsgebieten, wie zum Beispiel einer Namenswahl in einem Telefonbuch, keine großen Vorteile, da Namen schwierig erahnt werden können.

Mit dem **Benutzerkreis** sind die Anwender des Sprachsystems gemeint. In der *sprecherabhängigen Variante* beschränkt sich der Benutzerkreis auf nur eine bestimmte Person, wodurch das Sprachsystem auf die Sprechweise genau dieser Person eingestellt werden kann. Die *sprecherunabhängige Variante* hingegen hat das System Sprachsignale beliebiger Sprecher zu verarbeiten, was zu einer sehr hohen Variation der Sprechweisen führt. Meist wird in solchen Fällen ein *sprecheradaptives System* angewandt, welches sich durch eine Trainingsphase an die Sprechweise der Nutzer anpasst. Dies ist oft bei kommerziellen Tools der Fall.[24]

2.2.4 Fehlerarten der Spracherkennung

Leider zeigt die Praxis, dass in Spracherkennungssystemen immer wieder Fehler bei der Umwandlung passieren. Die Häufigkeit der Fehler unterscheidet somit die Qualität der Spracherkennungssysteme am Markt. Die Erkennungswahrscheinlichkeit ist eine Kenngröße, welche eine Art Bewertungsnote für Spracherkennungssoftware darstellt. In der Forschung wird sie für den Vergleich verschiedener Systeme und die Dokumentation der Verbesserungen herangezogen. Für die Bewertung werden drei Fehlerarten betrachtet:

1. Verwechslungen N_V:
   Ein falsches Wort wurde an der Stelle X erkannt.
2. Auslassungen N_A:
   An einer Stelle X wurde ein Wort nicht erkannt.
3. Einfügungen N_E:
   Ein zusätzliches Wort wurde an einer nicht geplanten Stelle X eingefügt.

Aus diesen drei Fehlerarten lassen sich zum einen die Wortkorrektheit (WK), welche nur Verwechslungen und Auslassungen betrachtet, und die Wortakkuratheit (WA), welche zusätzlich die Einfügungen betrachtet, berechnen.

[24] Vgl. Euler, S. (2006), S.18 ff.

Diese beiden Kenngrößen ergeben sich aus den folgenden Formeln, wobei N_W für die Anzahl der gesprochenen Wörter steht.[25]

$$Wortkorrektheit\ WK = \frac{N_W - N_V - N_A}{N_W}$$

$$Wortakkuratheit\ WA = \frac{N_W - N_V - N_A - N_E}{N_W}$$

2.3 Technische Funktionsweise

Allgemein bedeutet Spracherkennung, dass gesprochene Sprache in geschriebenen Text umgewandelt wird.[26] Dies geschieht mithilfe von mehreren statistischen Operatoren.[27] Der Schall wird auf mehreren Ebenen interpretiert. Diese Ebenen werden im Folgenden beschrieben.

2.3.1 Ebenen der Sprachverarbeitung

Die unteren Ebenen sind sprachsignalnah, während die oberen Ebenen intentionsnah sind.[28] Die unterste Ebene ist das Sprachsignal. Das ist prinzipiell das vom Menschen Gesprochene, das der Computer über das Mikrofon aufgenommen hat.

Auf der nächsthöheren Ebene befindet sich die Beobachtungsfolge. Um die Beobachtungsfolge feststellen zu können, muss das Sprachsignal zeitkontinuierlich und wertkontinuierlich diskretisiert werden. Das Signal wird zunächst mit einer bestimmten Abtastfrequenz zeitlich in diskrete Teilabschnitte unterteilt. Die Signalabschnitte sind sehr kurz und bewegen sich im Millisekunden Bereich. Anschließend werden die Signalabschnitte wertkontinuierlich unterteilt. Mit Hilfe des schnellen Fourier Transformation Algorithmus (englisch: Fast Fourier Transform) werden die Signalabschnitte analysiert.[29] Dieser Algorithmus ermittelt die Frequenzbereiche der Signalabschnitte.[30] Es werden mehrere Stützpunkte abgetastet, die entsprechend der Abstände der Mel-Skala gewählt werden.[31] Die Mel-Skala gibt die vom Menschen wahrgenommene Tonhöhe, die sogenannten Tonheit, an. Das Prinzip dieser Skala ist, dass die wahrgenommene Tonhöhe nicht linear zur Frequenz ist. Der Abstand zwischen zwei wahrgenommenen Tönen ist unabhängig von der absoluten Frequenz. Die Tonheit wird in sogenannten mel angegeben. Als Ausgangspunkt gilt, dass 1000 mel bei einer Frequenz

[25] Vgl. Euler, S. (2006), S.21 ff.
[26] Vgl. Oswald, P./Goebl, H./Brix, E. (2001) S. 249
[27] Vgl. Glaser, E. u. a. (2014), S. 207
[28] Vgl. Müller, J. (1997), S. 3
[29] Vgl. ebenda, S. 3 f.
[30] Vgl. Berger, P. (2005)
[31] Vgl. Müller, J. (1997), S. 4

von 1000 Hz liegen.[32] Die folgende Abbildung 1 zeigt wie sich die Tonheit und die Frequenz zueinander verhalten:

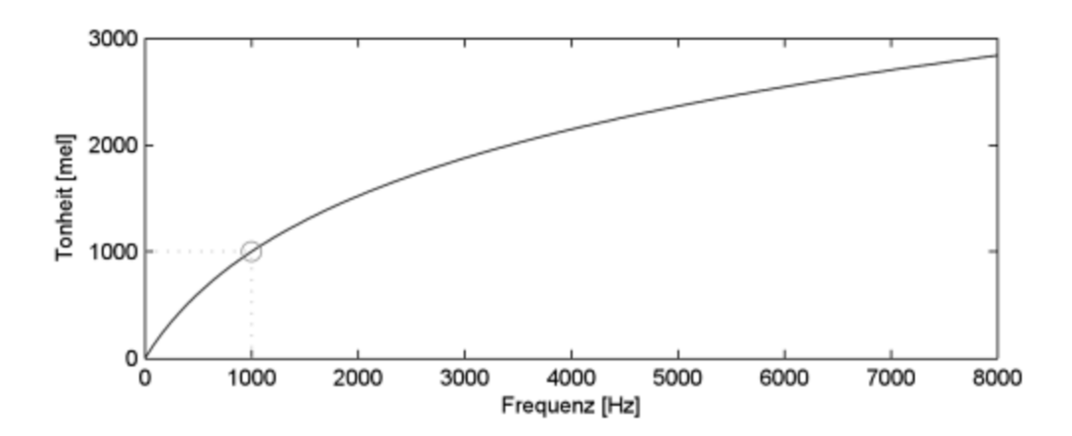


Abbildung 1: Zusammenhang Frequenzskala und Mel-Skala[33]

Durch den Einsatz der Mel-Skala kann die Frequenzauflösung des menschlichen Gehörs berücksichtigt werden. Nach der Feststellung der Stützpunkte des Signals mithilfe der Mel-Skala kann die sogenannte Beobachtungsfolge gewonnen werden. Durch die Stützpunkte können sogenannte Merkmalsvektoren gebildet werden. Diese Merkmalsvektoren beschreiben die Eigenschaften des Sprachsignals. Die Merkmalsvektoren werden sequentiell aneinandergereiht und bilden somit die Merkmalsvektorenfolge bzw. die Beobachtungsfolge.

Als nächstes folgt die Phonemebene, welche die phonetische Aussprache repräsentiert. Durch die Aneinanderreihung der ausgesprochenen Phoneme wird die Phonemkette gebildet.[34] Ein Phonem ist die „kleinste bedeutungsunterscheidende sprachliche Einheit“[35]. Zum Beispiel bei Bein und Pein unterscheiden sich nur die Phoneme “B“ und “P“.[36]

Über der Phonemebene liegt die Wortebene. Sie enthält eine Wortkette, die aus mehreren Wörtern besteht. Die Wortkette wurde durch die Anwendung von grammatischen und lexikalischen Regeln festgelegt.

Als nächstes folgt die semantisch-syntaktische Repräsentationsebene.[37] Dies ist die Zwischenebene zwischen der Wortebene und der Intention.[38] Der Bedeutungsinhalt der Äußerung wird durch die Analyse der Wortkette festgestellt und eine sogenannte semantische Gliederung wird identifiziert. Eine semantische Gliederung repräsentiert die inhaltliche Be-

[32] Vgl. Pfister, B./Kaufmann, T. (2008), S. 94 f.
[33] Enthalten in: ebenda, S. 95
[34] Vgl. Müller, J. (1997), S. 4
[35] Bibliographisches Institut GmbH (2013a)
[36] Vgl. Konradin Medien GmbH (o. J.)
[37] Vgl. Müller, J. (1997), S. 4
[38] Vgl. Krapichler, C. (1999), S. 54

deutung der gesprochenen Aussage. Die Identifikation der semantischen Gliederung bzw. der inhaltlichen Bedeutung einer Äußerung ist ein sehr komplexer Prozess, der auf einer probabilistischen Aussage über eine Wortkette basiert und mit Hilfe von stochastischen Algorithmen durchgeführt wird.[39] Ein stochastischer Algorithmus ist grob gesagt ein „Algorithmus, bei dem der Zufall mit ins Spiel kommt“[40]. Das bedeutet, dass der Zufall das letztendliche Ergebnis beeinflusst. Es sollen die Ergebnisse mit hoher Wahrscheinlichkeit ermittelt werden.[41] Für die semantische Gliederung bedeutet das, dass die inhaltliche Bedeutung mit der höchsten Wahrscheinlichkeit ermittelt werden soll. Mit Hilfe der semantischen Gliederung kann anschließend die Intention ermittelt werden.

Die Intention ist die oberste Ebene und stellt die „applikationsspezifischen Kommandos“[42] dar, welche die gewünschten Aktionen auslösen oder die gewünschten Informationen ausgeben. Die folgende Abbildung 2 zeigt eine Übersicht der eben beschriebenen Ebenen der Sprachverarbeitung:



Abbildung 2: Ebenen der Sprachverarbeitung

Die Sprachverarbeitung kann von oben nach unten, „Top-Down“[43] Ansatz, oder von unten nach oben, „Bottom-Up“[44] Ansatz, ablaufen. Da der Top-Down Ansatz in der Praxis am häufigsten eingesetzt wird, wird nur dieser Ansatz im Folgenden näher beschrieben.

[39] Vgl. Herbert Utz Verlag GmbH (o. J.)
[40] Eichelsbacher, P. (o. J.)
[41] Vgl. ebenda
[42] Müller, J. (1997), S. 5
[43] Ebenda, S. 5
[44] Ebenda, S. 5

2.3.2 Top-Down Ansatz

Beim Top-Down Ansatz beginnt die Sprachverarbeitung mit der Intention auf der obersten Ebene. Es wird eine große Anzahl an Intentionen generiert. Für jede Intention werden zugehörige semantische Gliederungen produziert. Aus den semantischen Gliederungen werden anschließend passende Wortketten gebildet. Für die Wortketten werden zugehörige Phonemketten und für diese entsprechende Beobachtungsfolgen generiert. Danach muss geprüft werden, ob das vorliegende Sprachsignal zu einer der Beobachtungsfolgen passt. Zuletzt muss die Entstehung der passenden Beobachtungsfolge zurückverfolgt werden, um die zugehörige Intention zu erhalten.[45]

Bei der Sprachverarbeitung werden unterschiedliche Modelle und Generatoren für die Überführung in die verschiedenen Ebenen genutzt. Die folgende Abbildung 3 stellt diese Modelle und Generatoren für die Top-Down Sprachverarbeitung dar:

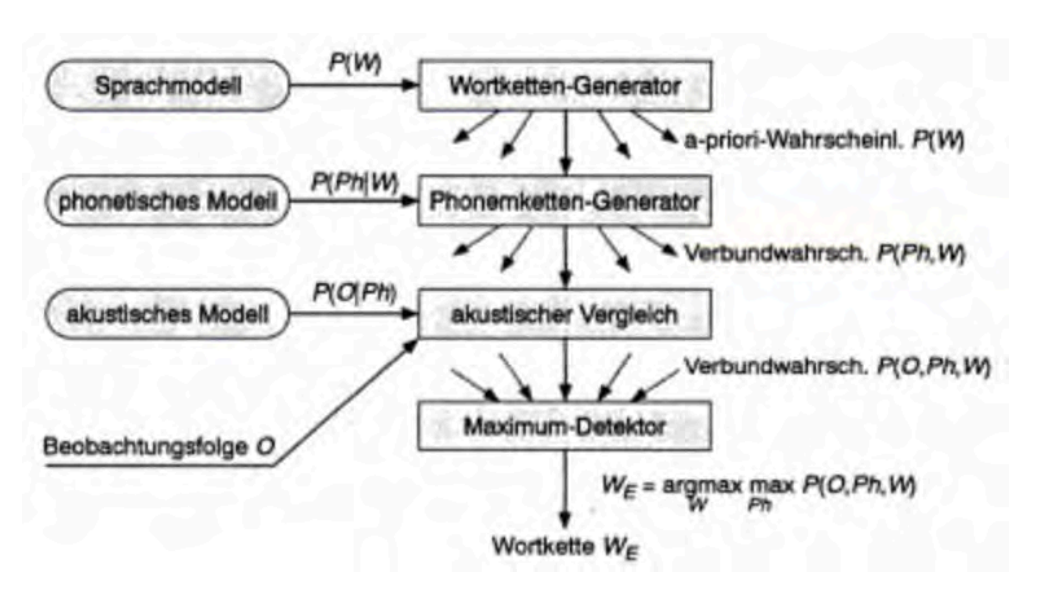


Abbildung 3: Top-Down Sprachverarbeitung[46]

Zunächst produziert der Wortkettengenerator mögliche Wortketten (W) und deren A-priori-Wahrscheinlichkeiten (P(W)). Mit Hilfe des Phonemketten-Generators werden für jede mögliche Wortkette die zugehörigen Phonemketten (Ph) mit der jeweiligen bedingten Wahrscheinlichkeit (P(Ph|W)) generiert. Anschließend folgt der akustische Vergleich. Hierbei wird die bedingte Wahrscheinlichkeit (P(O|Ph)) für jede generierte Phonemkette (Ph) und gegebene Beobachtungsfolge (O) identifiziert. Die Beobachtungsfolge stammt aus dem eingegebenen Sprachsignal. Letztlich ist die erkannte Wortkette (W_E) die Wortkette aus der Kombination einer Wortkette (W), einer Phonemkette (Ph) und der gegebenen Beobachtungsfolge (O) mit

[45] Vgl. Müller, J. (1997), S. 5 f.
[46] Enthalten in: ebenda, S. 17

der höchsten Wahrscheinlichkeit (P(O,Ph,W)). Die höchste Wahrscheinlichkeit wird mithilfe des Maximum-Detektors ermittelt.[47]

Die Wahrscheinlichkeiten werden durch Anwendung der verschiedenen Modelle ermittelt. Mithilfe des Sprachmodells wird die Auftrittswahrscheinlichkeit (P(W)) einer Wortkette (W) abgeschätzt. Eine Wortkette besteht aus NW Worten, das heißt: $W=w_1w_2...w_{NW}$. Das phonetische Modell enthält alle Worte und deren Aussprache, die dem Spracherkennungssystem bekannt sind. Es identifiziert die bedingte Wahrscheinlichkeit (P(Ph|W)) einer Phonemkette (Ph) bei einer vorhandenen Wortkette (W). Die Phonemkette besteht aus NP Phonemen, das heißt: $Ph=ph_1ph_2...ph_{NP}$. Es wird eine Phonemkette pro Wort gebildet. Anschließend hilft das akustische Modell dabei, die bedingte Wahrscheinlichkeit (P(O|Ph)) für das Auftreten einer Beobachtungsfolge (O) für eine gegebene Phonemkette (Ph) zu ermitteln. Eine Beobachtungsfolge besteht aus NO Merkmalsvektoren, das heißt: $O=o_1o_2...o_{NO}$.[48] Durch die Ermittlung der Wahrscheinlichkeiten kann die höchste Wahrscheinlichkeit und somit die erkannte Wortkette identifiziert werden. Somit kann das Gesprochene in eine Wortkette, also in schriftlichen Text umgewandelt werden.

Es ist sehr kompliziert, aus einer Phonemkette eine Beobachtungsfolge zu ermitteln. Ein möglicher Ansatz hierfür ist das Hidden-Markov-Modell (HMM). Das Modell wird im folgenden Kapitel dieser Arbeit beschrieben.

2.3.3 Das Hidden-Markov-Modell

Das Hidden-Markov-Modell wird in der Sprachverarbeitung genutzt, um zeitliche Auflösung der Sprache zu modellieren. Grundlage des HMMs ist das Markov-Modell, auch Markov-Prozess genannt.[49]

2.3.3.1 Markov-Modell/ Markov-Prozess

Bei dem Markov-Modell handelt es sich um einen speziellen stochastischen Prozess. Dieser Prozess berechnet zu jedem Zeitpunkt die Wahrscheinlichkeit aller zukünftigen Zustände, die vom momentanen Zustand abhängen.[50]

Ein solches Zufallsmodell entspricht einer zeitlichen Abfolge von messbaren Ereignissen. Diese Ereignisse werden als diskrete oder kontinuierliche Werte wahrgenommen, bei denen davon ausgegangen wird, dass ein Ereignis Einfluss auf das Nachfolgende hat und somit die

[47] Vgl. Müller, J. (1997), S. 17
[48] Vgl. ebenda, S. 18 f.
[49] Vgl. Wendemuth A. (2004), S.133
[50] Vgl. Petersen W. (2004)

Messwerte in ihrer zeitlichen Abfolge voneinander abhängig sind. Die Markov-Kette entspricht diesem Zufallsmodell. Hierbei handelt es sich um ein System, welches sich jederzeit in einem unterschiedlichen Zustand, S_1, S_2, … S_N, befindet. In einem bestimmten Intervall kann ein Zustandswechsel vollzogen werden. Die Wahrscheinlichkeiten eines nachfolgenden Zustandes hängt bei zeitlich aufeinanderfolgenden Ereignissen von dem vorausgehenden Zustand ab. D.h., eine geringe Anzahl an Zuständen bestimmt die probabilistische Zeitentwicklung eines Systems.[51]

2.3.3.2 Hidden-Markov-Modell

Wird der zu Grunde liegenden Zufallsprozesse nicht direkt beobachtet, spricht man vom Hidden-Markov-Modell. Beispielsweise wird ein Münzwurf betrachtet. Bei diesem Modell gibt es die Zustände „Kopf“ und „Zahl“. Im HMM wird eine Art Trennwart zwischen dem Münzenwerfenden und dem Beobachter aufgestellt. Der Münzwurf ist somit vor dem Beobachter versteckt. Dem Außenstehenden werden nur die beobachtungsfolge mitgeteilt, ihm ist nicht bekannt, welche und wie viele Münzen geworfen wurden. Ziel des HMMs ist es nicht die Anzahl der Münzen zu identifizieren, sondern es wird ein möglichst einfaches Modell gesucht, das die Situation ausreichend genau beschreibt. Beispielsweise werden drei Münzen geworfen, ist das Ziel des HMMs herauszufinden, welche Reihenfolge der jeweiligen Zustände vorkommt. Die Anzahl der durchgeführten Versuche wird hier nicht beachtet, sondern die Wahrscheinlichkeit, die der Münzwurf ergibt.[52]

2.3.3.3 Hidden-Markov-Modell in der Spracherkennung

In der Spracherkennung wird das wortbezogene HMM eingesetzt, jedoch besteht die Möglichkeit, phonembasierte Markov-Modelle zu nutzen. In dem phonembasiertem HMM wird für jedes Phonem Markov-Ketten gebildet.[53]

Das HMM wird in der Spracherkennung für die zeitliche Auflösung, um die Sprache zu modellieren, genutzt. Das HMM hat sich nicht nur in der Sprachverarbeitung durchgesetzt, sondern wird ebenfalls in allen Bereichen der Wirtschaft genutzt. Beispielsweise wird das HMM in der Bildverarbeitung oder Bioinformatik eingesetzt. Das HMM kann als doppeltes stochastisches Wahrscheinlichkeitsmodell aufgefasst werden (siehe Abbildung 4).[54]

[51] Vgl. Beer D. (o. J.)
[52] Vgl. ebenda
[53] Vgl. Wendemuth A. (2004), S.135
[54] Vgl. ebenda, S.135

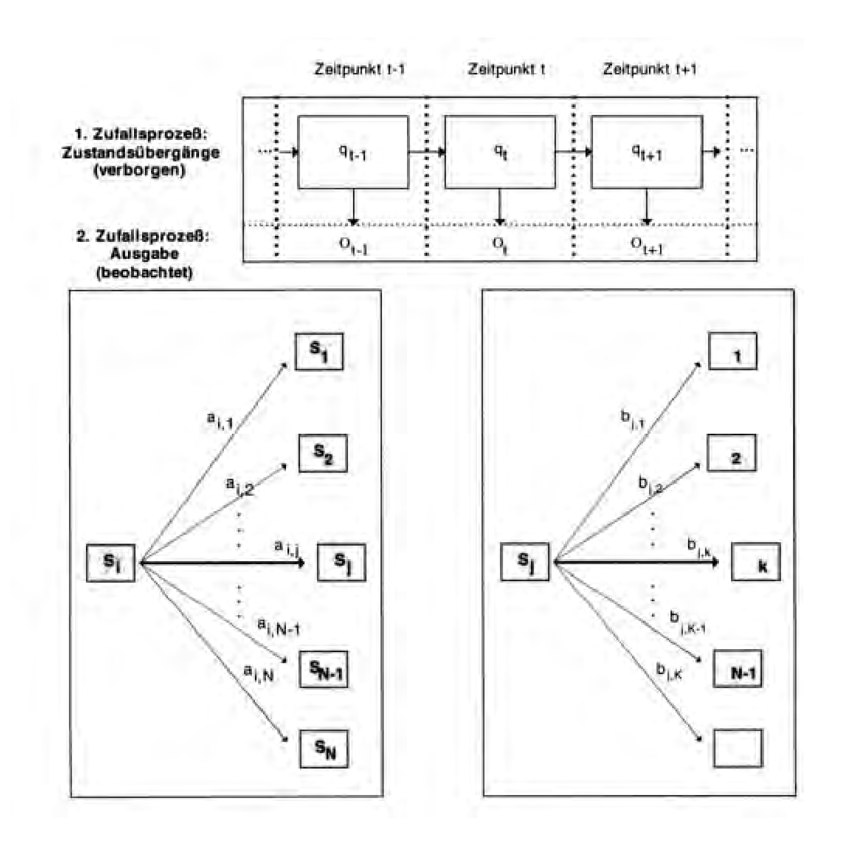


Abbildung 4: HMM als doppelter stochastischer Prozess[55]

Unterschieden wird zwischen dem 1. Zufallsprozess und dem 2. Zufallsprozess:[56]

1. Zufallsprozess:

   Hierbei handelt es sich um einen diskreten stochastischen Zufallsprozess, der sich auf die Zustandsübergänge bezieht. Es wird eine diskrete Zufallsfolge, q_1, q_2, …, q_n, erzeugt.

2. Zufallsprozess:

   Der zweite Prozess steuert die Ausgabe des Markov-Modells. Es wird eine Auswahl, nach der Ausgabewahrscheinlichkeit, eines zufälligen Zeichen des aktuellen States, getroffen.

Zu Beginn gibt es eine Trainingsphase in der bestimmte Phoneme oder Wortteile an das an den Sprecher angepasste HMM gespeichert wird. Diese Phoneme oder Wortteile werden Allophone genannt und enthalten bis zu acht Zustände. Jeder Zustand erhält außerdem Anfangs- oder Endwahrscheinlichkeiten. Die zeitlichen Variationen, die bei der Aussprache entstehen können, werden über eine Selbstreferenz hergestellt. Das heißt, ein langgesprochener Laut (Zustand) wird auf sich selbst abgebildet und dadurch im Modell verlängert. Für jedes Wort, auch Einheit genannt, wird mit dem HMM die Wahrscheinlichkeit berechnet, ob das gespeicherte Modell, bestehend aus einer Folge aus Zuständen, das aufgenommene

[55] Enthalten in: Wendemuth A. (2004), S.136
[56] Vgl. ebenda, S.135

Signal erzeugen kann. Diese Berechnung muss für jedes einzelne Modell durchgeführt werden. Dadurch kommt es zu einem hohen Rechenaufwand. Dieser Aufwand wird durch spezielle Rechenverfahren abgekürzt. Es entsteht eine ungenauere Berechnung die jedoch schneller durchgeführt wird. Beispielsweise wird auf folgende Algorithmen zurückgegriffen:

- Viterbi-Algorithmus
- Forward- Backward-Algorithmus
- Baum-Welch-Optimierungs-Regeln

Die Viterbi- und Forward-Backward-Algorithmen arbeiten mit Teilwahrscheinlichkeiten. Zuerst werden alle Wege berechnet, die das Ziel abbilden und dann der Wahrscheinlichste.

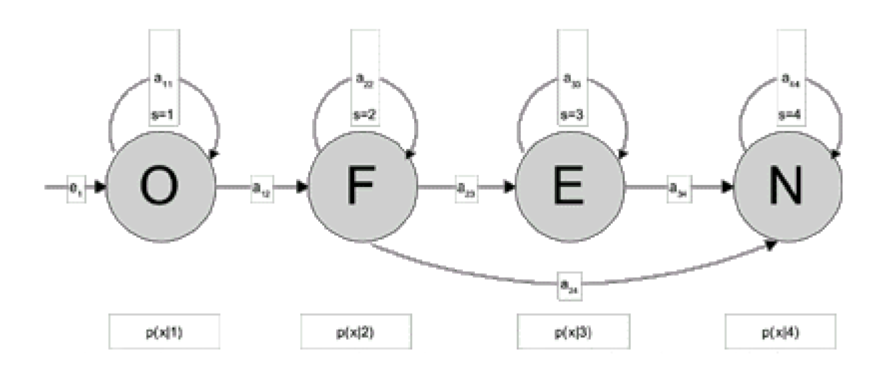


Abbildung 5: HMM am Beispiel "Ofen"[57]

Abbildung 5 zeigt ein Beispiel für ein HMM, in dem es für jeden Laut nur einen Zustand gibt. Bei dem Wort „Ofen“ wird beachtet, dass eine Verlängerung der einzelnen Vokale vorkommen kann. Dies wird durch die Selbstreferenz verdeutlicht. Ebenso besteht die Möglichkeit, dass bei Dialekten Buchstaben nicht ausgesprochen werden, beispielsweise das „e“ verschluckt wird und nur die Aussprache „Ofn“ entsteht.[58]

[57] Enthalten in: o. V. (o. J.a)
[58] Vgl. ebenda

3 Spracherkennungstools

Es gibt sowohl Open Source als auch kommerzielle Spracherkennungstools. In den folgenden Kapiteln 3.1 und 3.2 werden sowohl mögliche Open Source-Tools als auch kommerzielle Tools aufgezeigt und beschrieben.

3.1 Open Source-Tools

Es gibt viele verschiedene Open Source-Tools, die für die Spracherkennung genutzt werden können. Im Folgenden werden die Spracherkennungstools CMU Sphinx, Julius, RWTH ASR, Simon und VoxForge beschrieben.

3.1.1 CMU Sphinx

CMU Sphinx bietet verschiedene Reihen und Pakete, die verschiedene Aufgaben und Anwendungen für die Sprachanwendung zur Verfügung stellen. Unterschieden wird zwischen folgenden Tools:[59]

- Pocketsphinx – Sprachsteuerung - Recognizer in C geschrieben
- Sphinxbase – Unterstützende Bibliothek vorgesehen für den PocketSphinx
- Sphinx3 – Decoder für Spracherkennungssuche in C geschrieben
- Sphinx4 – Spracherkennung - Recognizer in Java geschrieben
- CMUclmtk – Sprachmodell Tools
- Sphinxtrain – Akustikmodell Trainingstools

Im Laufe des Projekts ist das Spracherkennungstool Sphinx 4 interessant, da Sphinx 3 eine ältere Vorgängerversion von Sphinx 4 ist. CMU Sphinx ist ein Projekt der Carnegie Mellon University das immer weiter entwickelt und ausgebaut wird. Es handelt sich hierbei um ein Open Source-Tool, das frei zur Verfügung steht.[60]

Die Spracherkennungssoftware Sphinx 4 basiert auf dem Wahrscheinlichkeitsprinzip des HMMs, welches in Kapitel 2.3.3 beschreiben wird. Des Weiteren kann Sphinx 4 mit dem Tool Sphinxtrain trainiert werden, d.h. Dialekte oder besondere Aussprachen werden angepasst.[61]

CMU Sphinx bietet eine deutsche Version des Spracherkennungstools Sphinx 4 an, die jedoch einen niedrigeren Entwicklungsstand der englischen Spracherkennungsversion besitzt. Ebenfalls wird die zukünftige Weiterentwicklung der deutschen Version nicht garantiert. [62]

[59] Vgl. CMU Sphinx (o. J.a)
[60] Vgl. CMU Sphinx (o. J.b)
[61] Vgl. CMU Sphinx (o. J.c)

Sphinx 4 kann von der Internetseite „cmusphinx.sourceforge.net“ heruntergeladen werden. Eine ausführliche Installationsanleitung ist auf der Internetseite vorhanden. Zur Nutzung werden keine weiteren Tools oder Vorinstallationen benötigt. Jedoch kommt es durch die Online-Verwaltung des Tools zur Komplikation. Der Rechner oder das Smartphone muss an eine Internetverbindung angeschlossen sein, damit die Software nutzbar ist. Ein Offline-Modus wird nicht angeboten.[63]

3.1.2 RWTH ASR

RWTH ASR (Aachen University Speech Recognition System) ist ein Softwarepaket, das neueste Spracherkennungstechnologien für die Entwicklung von akustischen Modellen und automatischen Spracherkennungssystemen beinhaltet. Seit 2001 wird es von der Human Language Technology and Pattern Recognition Group an der Rheinisch-Westfälischen Technischen Hochschule Aachen (RWTH) entwickelt.

Spracherkennungssysteme, die mit diesem Framework entwickelt wurden, wurden erfolgreich in mehreren internationalen Forschungs- und Entwicklungsprojekten verwendet. Das ist vor allem darauf zurückzuführen, da das Framework einen hohen Flexibilitätsgrad aufweist und sich leicht erweitern lässt. Denn RWTH ASR besteht aus mehreren Bibliotheken und Programmteilen, die in C++ geschrieben sind, die derzeit werden nur Linux (x86 und x86-64) und MAC OS X Plattformen unterstützten.[64] Allerdings enthält das Tool keinerlei akustische oder sprachliche Modelle. Diese müssen von externen Datenbanken bezogen werden.

Nachfolgend werden einige der vielfältigen Features erläutert.

Mit dem Spracherkennungsdecoder lassen sich große Mengen Vokabular kontinuierlicher Sprache erkennen. Dieser basiert auf Algorithmen, wie dem HMM, der wortabhängigen Baumsuche (mit der Unterstützung von Kreuzwort-Modellen) und Wortgittern. Die Emissionswahrscheinlichkeitsrechnung des HMM wurde dabei durch die Nutzung von Single Instruction Multiple Data -Befehlen optimiert. Mithilfe der Software lassen sich bei der Erkennung der Sprache verschiedene Merkmale extrahieren, dabei werden u.a. Vorgehensweisen wie Mel Frequency Cepstral Coefficients und Perceptual Linear Prediction eingesetzt, um das Frequenzspektrum darzustellen, und daraus die wahrgenommene Tonhöhe zu beschreiben.[65] Ferner werden mehrere Methoden verwendet, um den Spielraum einzugrenzen und die Sprache klarer zu erkennen, z.B.: Differenzierung zwischen mehreren Stimmen, Gewichtung der Grammatik, Vocal tract length normalization zur Unterscheidung zwischen

[62] Vgl. CMU Sphinx (o. J.d)
[63] Vgl. ebenda
[64] Vgl. RWTH ASR (2014)
[65] Vgl. Stadermann, Dipl.-Ing.J. (2005), S.7ff

maskuliner und femininer Stimme und Latent Drichlet allocation zur Themenerkennung. Mit der Zeit soll das System sich an den Sprecher anpassen. So reagiert es beispielsweise automatisch auf beliebte Redewendungen, Atempausen etc. (maximum likelihood linear regression). Zudem lassen sich Ein- und Ausgangsdaten leicht in XML oder Textformate umwandeln. Entsprechende Konvertierungstools sind enthalten.[66]

Um das Tool zu nutzen muss die RWTH ASR License, die von Q Public License abgeleitet wird, zugestimmt werden. Q Public License ist eine Softwarelizenz für Open Source-Software, die vom Unternehmen Trolltech erstellt wurde und die freie Nutzung im Sinne von Open Source garantiert. Die RWTH ASR License garantiert die kostenlose Nutzung, inklusive der Weiterverbreitung und Modifikation der Software für nicht kommerzielle Zwecke. Des Weiteren wird ein Set von installierten Tools und Bibliotheken zur Installation der aktuellsten Version 0.6.1, die auch als Demo-System inklusive akustischer Modelle verfügbar ist, vorausgesetzt.[67]

Für Fragen zur anschließenden Nutzung des Tools steht eine umfangreiche Dokumentation sämtlicher Softwarekomponenten und theoretischer Grundlagen bereit, für die jedoch teilweise eine Registrierung erforderlich ist. Sollten dennoch Anliegen offen sein, besteht die Möglichkeit diese im Support-Forum anzusprechen.[68]

3.1.3 Simon

Simon ist ein frei verfügbares Spracherkennungsprogramm. Ziel des Programms ist, die Bedienung mit Maus und Tastatur zu ersetzen.[69] So soll es beispielsweise Chats, E-Mails, Surfen im Internet und Online-Banking auch für körperbehinderte Personen und Senioren ermöglichen.[70] Das Programm ist so flexibel wie möglich gestaltet, sodass es mit unterschiedlichen Sprachen und Dialekten funktioniert.[71] Mögliche Sprachen sind Deutsch, Englisch und Spanisch.[72] Simon läuft auf den Betriebssystem von Windows und Linux.[73] Zudem wird aktuell an einer Version für Mac OS X gearbeitet.[74] Für Software-Entwickler kann der Source Code verwendet werden, um Simon selbstständig zu kompilieren und zu entwickeln. Neben Windows und Linux steht auch entsprechender Source Code für Mac OS X zur Verfügung.[75]

[66] Vgl. RWTH ASR (2014)
[67] Vgl. ebenda
[68] Vgl. ebenda
[69] Vgl. Grasch, P. (o. J.a)
[70] Vgl. Simon listens (2012c)
[71] Vgl. Grasch, P. (o. J.a)
[72] Vgl. Grasch, P. (o. J.b)
[73] Vgl. Grasch, P. (o. J.a)
[74] Vgl. Grasch, P. (2014)
[75] Vgl. KDE e.V. (2014)

Die Software beziehungsweise der Source Code kann von der Webseite von Simon (https://simon.kde.org) heruntergeladen werden.

Simon liegt eine Client-Server-Architektur zugrunde. Die Software Simon ist der Client. Es ist das grundsätzliche grafische Interface, das für den Endnutzer sichtbar ist. Simond ist wiederum der Server, auf dem die eigentliche Erkennung läuft.[76] Auf Simond wird auch das Sprachmodell verwaltet.[77] Das eingesetzte Sprachmodell ist ein HMM. Die dritte Komponente ist KSimond. Das ist das grafische Front-End des Servers Simond.[78]

Simon funktioniert mit CMU Sphinx und/ oder mit einer Kombination vom Hidden Markov Toolkit (HTK) und von Julius.[79] CMU Sphinx wurde bereits in Kapitel 3.1.1 beschrieben. HTK ist ein Werkzeugsatz, mit dem HMM gebaut und manipuliert werden können.[80] Julius ist eine sogenannte „large vocabulary continuous speech recognition"[81] Dekodierungssoftware. Es ist eine Spracherkennungssoftware für Diktate und großen Wortschatz. Durch vorhandene Schnittstellen kann es mit Standards wie die HMMs verwendet werden.[82] Julius wird zudem in Kapitel 3.1.5 näher beschrieben.

Simon bietet unterschiedliche Funktionalitäten. Zum einen können Vokabeln hinzugefügt und entfernt werden. Zum anderen kann Grammatik importiert werden. Des Weiteren kann Simon trainiert werden, um die Spracherkennung zu verbessern.[83] Außerdem gibt es viele weitere Funktionen, die sogenannten Kommandos. Die Kommandos umfassen unter anderem die Folgenden:[84]

- Programm starten,
- Öffnung einer URL, wie zum Beispiel eines Ordners,
- Verknüpfung eines Kommandos mit einem Text durch Text-Makros,
- Steuerung der Computermaus,
- Diktat,
- Künstliche Intelligenz für Konversationen mit der Software,
- Taschenrechner,
- Virtuelle, sprachgesteuerte Tastatur,
- Sprachausgabe.

[76] Vgl. Grasch, P. (o. J.b)
[77] Vgl. Simon listens (2012a)
[78] Vgl. Grasch, P. (o. J.b)
[79] Vgl. Grasch, P. (o. J.a)
[80] Vgl. o. V. (o. J.b)
[81] Julius development team (2014a)
[82] Vgl. Simon listens (2012b)
[83] Vgl. Grasch, P. (o. J.b)
[84] Vgl. ebenda

Simon bietet viele verschiedene Funktionen. Zudem bietet das Programm eigene Entwicklungsmöglichkeiten durch den Source Code, der frei verfügbar ist. Aktuell wird Simon ebenso weiterentwickelt. Daraus können sich zukünftig weitere Möglichkeiten für den Einsatz von Simon entwickeln.

3.1.4 VoxForge

VoxForge ist weniger ein Tool um Sprache zu erkennen, sondern vielmehr ein frei zugänglicher Aufbewahrungsort zur Sammlung von transkribierter Sprache.[85] Bei transkribierter Sprache handelt es sich um gesprochene Sprache, welche in schriftliche, „mit lautlich ungefähr entsprechenden Zeichen des lateinischen Alphabets"[86] entsprechenden Zeichenkombinationen umgewandelt werden.[87] Jeder Besucher der Internetseite kann zu dem Projekt beitragen. Hierzu wird eine Webapplikation zur Verfügung gestellt, durch welche die Besucher vorgegebene Textabschnitte vorlesen können. Dies kann völlig anonym durchgeführt werden. Lediglich Angaben wie Geschlecht, Altersgruppe, Aussprachevarietät und Mikrophontyp müssen angegeben werden.[88] Alle durch die Aufnahmen gewonnen Sprachmodelle werden für freie und quelloffene Spracherkennungssysteme zur Verfügung gestellt. Diese stehen unter der GNU General Public License[89], welche Entwicklern die Freiheit zur Teilung und Abänderung der Daten gibt.[90] Unter den Sprachmodellen sind das akustische Modell und das Grammar Modell beziehungsweise das Language Modell zu verstehen, welche beide für Spracherkennungstools benötigt werden. Das akustische Modell ist ein transkribiertes Modell, welches aus einer Vielzahl von Aufnahmen erzeugt wurde. Das Grammar Modell und das Language Modell haben dieselbe Funktion, unterscheiden sich jedoch in ihrem Umfang. Das Grammar Model ist lediglich eine sehr kleine Sammlung von Wortkombinationen aus denen hervorgeht, welche Wörter in welcher Kombination vorkommen können. Es eignet sich zum Beispiel besonders für die Gerätesteuerung, für die nur bestimmte Wörter, wie „Start" und „Stopp", benötigt werden. Das Language Model ist eine deutlich größere Version. Es enthält die Wahrscheinlichkeiten des Aufeinanderfolgens von Wörtern und wird zum Beispiel bei der Erkennung längerer Texte verwendet.[91]

[85] Vgl. o. V. (o. J.d)
[86] Bibliographisches Institut GmbH (2013d)
[87] Vgl. ebenda
[88] Vgl. o. V. (o. J.c)
[89] Vgl. o. V. (o. J.d)
[90] Vgl. o. V. (o. J.e) – ohne jegliche Gewähr oder Garantie
[91] Vgl. o. V. (o. J.f)

3.1.5 Julius

Julius ist ein Spracherkennungs-Tool für Diktate und große Vokabelstämme, eine Large Vocabulary Continious Speech Recognition Software. Das heißt, Julius ist ein Tool, das ununterbrochene Sprachflüsse versteht und mit großen Wortschätzen umgehen kann.[92]

Julius basiert auf dem in Kapitel 2.3.3 erklärten HMM Modell, dem DFA Grammatiken und N-Gram Modellen. Es läuft hauptsächlich auf dem Betriebssystem Linux, kann jedoch auch auf Windows-Rechnern installiert werden. Julius ist ein Open Source Tool und kann frei ohne Kosten genutzt werden.[93]

Ursprünglich wurde Julius als japanisches Spracherkennungstool entwickelt, jedoch wurde die Software weiterentwickelt und kann sowohl für die deutsche als auch englische Sprache genutzt werden. Das Spracherkennungstool Simon ist die deutschsprachige Weiterentwicklung.[94]

Auf dieses Spracherkennungs-Tool wird aufgrund der Einschränkung auf die Sprache Japanisch nicht weiter eingegangen. Die deutschsprachige Weiterentwicklung Simon wird im Kapitel 3.1.3 erläutert.

3.2 Kommerzielle Tools

Es gibt verschiedene kommerzielle Spracherkennungstools, die heutzutage zum Einsatz kommen. Marktführer der kommerziellen Spracherkennungstools sind die Tools von Nuance Communications. Im folgenden Kapitel wird deshalb das Spracherkennungstool Nuance Dragon NaturallySpeaking beschrieben.

3.2.1 Nuance Dragon NaturallySpeaking 13

Dragon NaturallySpeaking ist eine kommerzielle Spracherkennungssoftware, die gesprochene Sprache sowohl in Text, als auch in Sprachbefehle auf dem Computer umwandelt. Dragon kann für private oder unternehmerische Zwecke verwendet werden. Aktuell steht Version 13 ab 99 Euro für private Anwender bereit. Die fünf wichtigsten Aufgabenfelder sind Dokumente erstellen, im Internet suchen, E-Mail und Kalender verwalten, Nutzung Sozialer Netzwerke und die Diktierfunktion. Schon die private Version bietet eine Fülle an Sprachbefehlen, wie z.B. „Starte Microsoft Word“ oder „Mach die vorherige Zeile fett“, um die Arbeit

[92] Vgl. Simon listens (2012b)
[93] Vgl. Julius development team (2014b)
[94] Vgl. Simon listens (2012b)

am Computer zu erleichtern.[95] Mittlerweile bietet Nuance eine Vielzahl an Varianten für verschiedenste Einsatzbereiche und Branchen (z.B. mobile Anwendungen, Barrierefreiheit, Kundendienst, Energie und Versorgung, Versicherungen etc.). Auf Letzteres wird später etwas genauer eingegangen.

Dragon Systems wurde 1982 von James und Janet Baker mit dem Ziel der Kommerzialisierung der Spracherkennungstechnologie gegründet. 1997 erschien die erste Version von Dragon NaturallySpeaking, die erstmals kontinuierliche Spracherkennung ermöglichte. Über die Jahre wurde das System kontinuierlich weiterentwickelt, sodass das heutige Unternehmen Nuance Communications Inc. Marktführer von kommerziellen Spracherkennungstechnologien ist.[96]

Dragon NaturallySpeaking bietet eine komplette Spracherkennungslösung und deckt von der herkömmlichen Diktierfunktion bis hin zur Sprachsteuerung sämtliche Spracherkennungsfunktionen ab. Es ist ein sprecherabhängiges System, bei dem der über ein Mikrofon gesprochene Text unmittelbar vom Rechner umgesetzt wird nach dem Prinzip „what you say is what you see“. Anders als mobile Lösungen von Smartphoneherstellern, ergibt sich dadurch eine schnellere und exaktere Umsetzung des Gesprochenen sowie die Möglichkeit zur Anpassung an das Nutzungsverhalten, da dieses nicht auf einem zentralen Server verarbeitet wird. Das System tastet akustische Signale ab und ordnet ihnen verschiedene Eigenschaften zu, wodurch eine Zuordnung von Lauten ermöglicht wird. Die Umsetzung basiert auf Varianten des HMM. Mit fortlaufender Nutzung (z.B. Korrektur von Erkennungsfehlern) werden die akustischen Modelle an den Sprecher angepasst. Das System berechnet, wie wahrscheinlich es ist das ein bestimmtes Wort gesagt wurde und in den Kontext passt. Am besten ist die Genauigkeit, wenn mehrere Wortgruppen, oder auch ganze Sätze, hintereinander gesprochen werden, da das System ausschließlich auf statistischen Methoden beruht und nicht auf grammatikalischen Regeln. Daher ist die Software vor allem für Diktate geeignet und nicht für die Umsetzung von Gesprächen.[97]

Bei Auslieferung der Software wird bei der privaten Version ein Vokabular von ca. 150.000 Wortformen geliefert, welches durch Korrekturen des Nutzers erweitert wird. Jedoch ist es nicht möglich Sprachen zu mischen, die Sprachmodelle der Software sind bis auf gängige Fremdwörter nur auf eine Sprache ausgerichtet.[98]

Um Dragon nutzen zu können wird ein Windows-System (z.B. Windows 7, 8 oder Windows Server 2012) vorausgesetzt. Eine Version für MAC OS X existiert ebenfalls, die jedoch bei weitem nicht so umfangreich ist wie die Windows-Version. Systeme mit Linux-Kern werden

[95] Vgl. Nuance Communications Inc. (o.J.)
[96] Vgl. Maher, T./ Arciniega, J. (o.J.)
[97] Vgl. Riedel, J. u.a. (2014)
[98] Vgl. ebenda

generell nicht unterstützt.[99] Um die Erkennung der Sprache zu ermöglichen wird ein Mikrofon vorausgesetzt.

Die Installation der Software soll relativ zügig von statten gehen. Allerdings kommt die Dauer auch auf die Anzahl der selektierten Sprachpakete an. Danach erwartet den Anwender zum ersten Programmstart ein Einführungsassistent, der bei der Einrichtung der Software assistiert. Dabei besteht die Möglichkeit mehrere Nutzerprofile anzulegen, die nachher die Individualisierung des Vokabulars ermöglichen. Im weiteren Verlauf wird das Mikrofon durch das Vorlesen von kurzen Texten in Lautstärke und Qualität justiert. Zudem ist es möglich dem Programm Zugriff auf Mails und Dokumente zu gewähren, damit dieses von Fachvokabular profitiert.[100]

Nach dem Start von Dragon NaturallySpeaking steht dem Nutzer eine simple, aber dennoch intuitiv bedienbare Oberfläche zur Verfügung, die sich leicht selbst erklärt. Sie beschränkt sich auf das Nötigste, wie z.B. Aufnahmestart, Modi-Wechsel, etc., und lässt viel Freiraum zur Bedienung der Textbearbeitungsprogramme.[101]

Wie eingangs erwähnt bietet Nuance die Software auch für spezifische Branchen, darunter Dragon NaturallySpeaking für das Versicherungswesen. Der Außendienst- oder interne Sachbearbeiter wird effektiv bei der Erstellung von Schadensfällen unterstützt. Dies äußert sich dadurch, dass er die Zeit zum Eintippen der Fälle spart und so eine höhere Quote erfüllen kann. Zudem stehen u.a. Funktionen zum sprachgesteuerten Erstellen, Senden und Verwalten von Emails bereit.[102] Dabei wird für jeden Mitarbeiter ein individuelles Benutzerprofil angelegt, das benutzerspezifische Makros, angepasstes Vokabular, sowie Informationen über Sprech- und Ausdrucksweise enthält. Mit dem Einsatz dieser professionellen Version und vor allem durch das angepasste fachspezifische Vokabular soll sich die Anzahl der Fehler um bis zu 30 % reduzieren.[103]

Verantwortlich für den Erfolg der Spracherkennungssoftware von Nuance zeigt sich die geringe Fehlerquote, die auch von sämtlichen Testzeitschriften honoriert wird. 99 % Genauigkeit wird im Durchschnitt erreicht.[104] Generell gilt, dass wie eben schon erwähnt Fachvokabular sich nochmals positiv auf die Genauigkeit auswirkt. Probleme sollen jedoch gelegentlich bei zusammengesetzten Substantiven auftreten. Daher gilt der Grundsatz auch hier, dass der Anwender das Ergebnis der Spracherkennung nochmals kontrollieren sollte.

[99] Vgl. Nuance Communications Inc. (o.J.b)
[100] Vgl. o.V. (o.J.g)
[101] Vgl. ebenda
[102] Vgl. Nuance Communications Inc. (o.J.c)
[103] Vgl. Kronauer Software (o.J.)
[104] Vgl. Spehr, M. (2014)

3.3 Kriterienkatalog

Ein Kriterienkatalog ist eine „Geschlossene Darstellung von Anforderungen an Produkte und Systeme“[105]. Mittels des Kriterienkatalogs werden die Spracherkennungstools getestet und bewertet. Je nach Anforderungen werden die Kriterien unterschiedlich gewichtet. Beispielsweise ist das Kriterium, dass Fachvokabular in dem Tool beinhaltet ist oder hinzugefügt werden kann wichtiger, als dass verschiedene Dialekte erkannt werden.

3.3.1 Allgemeines

Der Kriterienkatalog wurde auf Basis eigenerstellter Kriterien zusammengestellt. Die Gewichtung der Kriterien ist prozentual dargestellt. Alle Kriterien zusammengezählt ergeben 100%, d.h. je höher der prozentuale Anteil eines einzelnen Kriteriums, desto höher die Bedeutung in der Gesamtauswertung. Für die Bewertung der Kriterien wurde eine Benotung von 1 bis 5 gewählt. 5 bedeutet, das Kriterium ist in diesem Tool sehr gut gegeben, 1 bedeutet, das Kriterium wird nicht erfüllt. Beispielsweise dauert die Umwandlung eines gesprochenen Textes bestehend aus 5 Wörtern über eine Minute, wird die Note 1 vergeben. Dauert die Umwandlung nur 4-5 Sekunden wird eine sehr gute Bewertung gewählt, die Note 5. Die Auswertung des Kriterienkatalogs dient als Grundlage für die Entscheidung, welche Tools genutzt werden können und ob es sich lohnt diese Tools in späterer Ausarbeitung näher zu betrachten.

In folgender wird die Beschreibung und Gewichtung der Kriterien des Kriterienkatalogs dargestellt:

Kriterium	Beschreibung	Gewichtung (Begründung)
Einsatzgebiete	Besagt, in welchem Bereich das Spracherkennungstool typischerweise eingesetzt wird. Dieses Kriterium kann nicht bewertet werden. Es sollen nur die standardmäßigen Einsatzgebiete des Tools aufzeigen.	Da dieses Kriterium nicht bewertet wird, ist es nicht in der Gesamtbewertung enthalten. Es ist nur informativ. → Gewichtung: **0%**
Funktionalitäten	Es werden die Funktionalitäten des Spracherkennungstools aufgelistet. Die Bewertung erfolgt nach den Funktionalitäten. Diese wurden je nach Relevanz folgendermaßen sortiert:	Es ist relativ wichtig, inwiefern zusätzliche Funktionalitäten vorhanden sind. Beispielsweise durch das Hinzufügen von Vokabeln oder

[105] gbt.ch (o. J.)

	5 = Hinzufügen von Vokabeln 4 = Training eines Tools zur besseren Spracherkennung 3 = Kommandos (z.B. Aufruf von anderen Programmen) 2 = sonstige Funktionalitäten 1 = keine zusätzlichen Funktionalitäten	Training des Tools kann die Effektivität und Effizienz des Tools gesteigert werden. → Gewichtung: **4%**
Vokabular – Anzahl der Sprachen	Das besagt, wie viele verschiedene Sprachen vom Tool erkannt werden. Die Bewertung erfolgt folgendermaßen: 5 = Mehr als 4 Sprachen 4 = 4 Sprachen 3 = 3 Sprachen 2 = 2 Sprachen 1 = 1 Sprache	Die Bewertung der Anzahl an Sprachen ist nicht so wichtig. Es wesentlich wichtiger, welche Sprachen erkannt werden. → Gewichtung: **2%**
Vokabular – Sprachen	Dieses Kriterium besagt, welche Sprachen vom Tool erkannt werden. Die Bewertung über die Relevanz der Sprachen für den Einsatzzweck. Die Relevanz ist folgendermaßen: 5 = Deutsch 4 = Englisch 3 = Französisch, Italienisch, Portugiesisch, Spanisch 2 = Russisch, Chinesisch (Mandarin) 1 = andere Sprachen Für die Bewertung gilt die vorhandene Sprache mit höchster Relevanz. Die Auswertung lässt sich am folgenden Beispiel erklären:	Je nach Sprache kann die Bewertung unterschiedlich ausfallen. Wenn die Sprache Deutsch ist, ist die Bewertung hoch, da diese für den Einsatzzweck gebraucht werden. Falls die Sprachen nur Japanisch, Russisch und Chinesisch (Mandarin) umfassen, ist die Bewertung niedrig, da sich diese Sprachen nicht für den Einsatzzweck eignen. → Gewichtung: **7%**

	Ein Spracherkennungstool enthält die Sprachen Deutsch und Französisch. Das Kriterium wird mit 5 bewertet.	
Vokabular – Wortschatz/ Themenbereiche	Das Kriterium bewertet, welche Themenbereiche standardmäßig im Vokabular des Spracherkennungstools vorhanden sind. Die Bewertung erfolgt folgendermaßen: 5 = allgemeiner und fachspezifischer Wortschatz 4 = *Keine Wertung bei diesem Kriterium.* 3 = allgemeiner Wortschatz 2 = *Keine Wertung bei diesem Kriterium.* 1 = nur fachspezifischer Wortschatz	Welche Themenbereiche vorhanden sind, ist für den Einsatzzweck des Spracherkennungstools nicht so wichtig. Fehlende Themenbereiche können ggf. durch das Hinzufügen von Vokabeln ausgeglichen werden. → Gewichtung: **3%**
Vokabular - Umfang	Das Kriterium bewertet, wie umfangreich das Vokabular des Spracherkennungstools standardmäßig ist. Da bei den meisten Tools keine Informationen für die Prüfung dieses Kriteriums zu Verfügung stehen, kann das Kriterium nicht bewertet werden. Insofern Informationen zur Verfügung stehen, werden diese informativ aufgezeigt.	Da bei den meisten Tools keine Informationen für die Prüfung dieses Kriteriums zu Verfügung stehen, kann das Kriterium nicht bewertet werden. Das Kriterium ist nicht in der Gesamtbewertung enthalten. → Gewichtung: **0%**
Vokabular – Import	Das Kriterium beschreibt, inwiefern Wörter zu dem Vokabular des Tools hinzugefügt werden können, um das Vokabular zu erweitern. Die Bewertung erfolgt folgendermaßen: 5 = Import von Vokabular möglich 4 = *Keine Wertung bei diesem Kriterium.* 3 = *Keine Wertung bei diesem Kriterium.* 2 = *Keine Wertung bei diesem Kriterium.*	Das Vorhandensein einer Importfunktion ist relativ wichtig. Wenn keine Importfunktion vorhanden ist, hat das zwar keine negative Auswirkung auf die Effektivität des Spracherkennungstools, aber durch Imports kann das Vokabular erweitert werden. Das heißt, dass auch Fach-

	1 = Nicht möglich	begriffe und ähnliches eingefügt und somit erkannt werden können. → Gewichtung: **6%**
Kompatibilität – Endgeräte	Es wird bewertet, auf welchen Endgeräten das Spracherkennungstool funktioniert. Die Bewertung erfolgt folgendermaßen: 5 = Computer, Tablet-PC und Smartphone (alle 3) 4 = *Keine Wertung bei diesem Kriterium.* 3 = Computer und Tablet-PC, Computer und Smartphone oder Tablet-PC und Smartphone (2 von 3) 2 = *Keine Wertung bei diesem Kriterium.* 1 = Computer, Tablet-PC oder Smartphones (1 von 3)	Die Kompatibilität mit verschiedenen Endgeräten ist wichtig für die Nutzung des Spracherkennungstools. Umso mehr Endgeräte kompatibel sind, desto flexibler ist das Spracherkennungstool einsetzbar. → Gewichtung: **6%**
Kompatibilität – Betriebssysteme	Es wird betrachtet, auf welchen Betriebssystemen das Spracherkennungstool läuft. Die Bewertung erfolgt entsprechend der Betriebssysteme folgendermaßen: 5 = Aktuelle Betriebssysteme von Windows, Apple und Linux 4 = Aktuelle Betriebssysteme von Windows und Apple 3 = Aktuelle Betriebssysteme von Windows oder Apple 2 = Aktuelle Betriebssysteme von Linux 1 = Keine aktuellen Betriebssysteme Da Linux weniger verbreitet ist als die Betriebssysteme von Windows und App-	Die Kompatibilität mit verschiedenen Betriebssystemen ist wichtig. Nur wenn das Spracherkennungstool auf aktuellen Betriebssystemen läuft, ist es relevant für den Einsatzzweck. → Gewichtung: **6%**

	le, ist es für die Bewertung weniger relevant.	
Kompatibilität – Programmiersprachen	Es wird betrachtet mit welchen Programmiersprachen das Tool kompatibel ist. Die Bewertung erfolgt folgendermaßen: 5 = mit allen Programmiersprachen kompatibel 4 = mit Java, C++, Java Script und PHP 3 = mit 3 der angegebenen Programmiersprachen 2 = mit 2 der angegebenen Programmiersprachen 1 = mit einer der angegebenen Programmiersprachen	Die Kompatibilität mit mehreren Programmiersprachen ist wichtig. Nur wenn ein Spracherkennungstool sich mit aktuellen Programmiersprachen programmieren lässt, kann es ohne Einschränkungen genutzt werden. → Gewichtung: **4%**
Technische Anforderungen	Technische Anforderungen umfassen beispielsweise Prozessor, Speicherplatz und Arbeitsspeicher.	Da diese Informationen bei den meisten Tools nicht angegeben sind, wird dieses Kriterium nicht in die Gesamtbewertung mit einbezogen. Dieses Kriterium dient als Information. → Gewichtung: **0%**
Offline-Modus	Es wird betrachtet, ob die Möglichkeit besteht das Spracherkennungstool offline zu nutzen. Die Bewertung erfolgt folgendermaßen: 5 = Offline 4 = *keine Wertung bei diesem Kriterium* 3 = *keine Wertung bei diesem Kriterium* 2 = *keine Wertung bei diesem Kriterium* 1 = nur Online	Die Möglichkeit ein Spracherkennungstool Offline zu nutzten ist nicht so wichtig wie die Kompatibilität mit den aktuellen Betreibsystemen. Heutzutage besteht die Möglichkeit an fast jedem Ort Internet zu empfangen und damit ist es auch fast überall möglich Online-Spracherkennungstools zu

		nutzen. → Gewichtung: **5%**
Installations-aufwand	Es wird betrachtet, wie aufwendig die Installation des Tools ist. Unterschieden wird zwischen geringerem Aufwand und hohem Aufwand. Die Bewertung erfolgt folgendermaßen: 5 = schnelle und einfache Installation 4 = einfache aber aufwendige Installation 3 = schwierige aber schnelle Installation 2 = schwieriger und hoher Installations-aufwand 1 = Installation ohne IT-Experten nicht möglich	Wie hoch der Aufwand der Installation des Tools ist, ist für die Bewertung des Tools nicht so wichtig. Besteht ein höherer Aufwand kann die Installation von der IT des Unternehmens durchgeführt werden. → Gewichtung: **6%**
Registrierung	In diesem Kriterium wird betrachtet, ob sich der Nutzer des Tools sich Registrie-ren muss. Die Bewertung erfolgt folgendermaßen: 5 = keine Registrierung nötig 4 = *keine Wertung bei diesem Kriterium* 3 = einmalige Registrierung zum gren-zenlosen Erwerb des Tools 2 = *keine Wertung bei diesem Kriterium* 1 = Registrierung für jeden Nutzer muss erfolgen	Ob eine Registration benötigt wird, ist ein wichtiger Aspekt. Betrachtet wird, ob jeder ein-zelne Nutzer sich anmelden muss oder ob eine einfache Anmeldung nur für den Er-werb des Tools benötigt wird. → Gewichtung: **5%**
Sprecherab-hängigkeit	Sprecherabhängigkeit bedeutet, dass jeder Nutzer das Spracherkennungstool trainieren muss, bevor es genutzt werden kann. Somit ist das Tool speziell auf den Nutzer konfiguriert und kann nur von diesem Nutzer genutzt werden. Die Bewertung erfolgt folgendermaßen:	Die Sprecherabhängigkeit ist ein wichtiges Kriterium. Ist das Spracherkennungstool sprecherabhängig, muss es für jeden Nutzer einzeln trai-niert werden. Dies ist vor allem im Hinblick auf den

	5 = Sprecherunabhängiges System 4 = *Keine Wertung bei diesem Kriterium.* 3 = *Keine Wertung bei diesem Kriterium.* 2 = *Keine Wertung bei diesem Kriterium.* 1 = Sprecherabhängiges System	flexiblen Einsatzzweck mit vielen verschiedenen Nutzern nicht sehr nützlich. → Gewichtung: **8%**
Sprachadaptives System	Ein sprachadaptives System lernt mit der Zeit dazu. Es lernt bei der Benutzung des Spracherkennungstools durch den Nutzer. So können Sprachgewohnheiten und häufig benutzte Sätze einfacher identifiziert und umgewandelt werden. Die Bewertung erfolgt folgendermaßen: 5 = Ist ein sprachadaptives System 4 = *Keine Wertung bei diesem Kriterium.* 3 = *Keine Wertung bei diesem Kriterium.* 2 = *Keine Wertung bei diesem Kriterium.* 1 = Kein sprachadaptives System	Ob das System sprachadaptiv ist oder nicht, ist nicht sehr wichtig. Es ist ein praktisches Feature vor allem in Hinsicht auf die Effizienz des Spracherkennungstools. Allerdings ist es nicht notwendig dafür, dass die Spracherkennung korrekt funktioniert. → Gewichtung: **8%**
Dokumentation	Dieses Kriterium betrachtet, ob Handbücher, Tutorials etc. vorhanden sind. Die Bewertung erfolgt folgendermaßen: 5 = Dokumentation über Bedienung und technische Handbücher vorhanden. 4 = Bedienungshandbücher vorhanden 3 = Technische Handbücher vorhanden 2 = *Keine Wertung bei diesem Kriterium.* 1 = Keine Dokumentationen vorhanden	Das Vorhandensein von Dokumentationen ist relativ wichtig für die Nutzung und ggf. Implementierung des Spracherkennungstools. → Gewichtung: **8%**
Datenschutz	Werden die Daten verschlüsselt übertragen und ggf. verschlüsselt gespeichert? Die Bewertung erfolgt folgendermaßen: 5 = Datenschutz bei Übertragung und	Der Datenschutz ist ein wichtiges Kriterium. Daten die umgewandelt und ggf. transferiert werden, müssen durch gewisse Maßnahmen ge-

	Speicherung 4 = *Keine Wertung bei diesem Kriterium.* 3 = *Keine Wertung bei diesem Kriterium.* 2 = *Keine Wertung bei diesem Kriterium.* 1 = Kein Datenschutz	schützt sein. → Gewichtung: **8%**
Lizenzen	Werden Lizenzen benötigt? Welches Lizenzmodell steht zur Verfügung (Lizenz für das gesamte Unternehmen, Lizenz je Nutzer)? Die Bewertung erfolgt folgendermaßen: 5 = Keine Lizenzen notwendig 4 = *Keine Wertung bei diesem Kriterium.* 3 = Lizenz für das gesamte Unterneh-men 2 = *Keine Wertung bei diesem Kriterium.* 1 = Lizenz je Nutzer	Die Betrachtung der Lizenz-anforderungen ist wichtig für die Einsetzbarkeit und die Kosten des Spracherken-nungstools. Wird eine Lizenz je Nutzer gebraucht, kann es sein, dass die Kosten relativ hoch sind. Zudem kann das Tool nicht flexibel eingesetzt werden, da für jeden Nutzer eine eigene Lizenz benötigt wird. → Gewichtung: **6%**
Kosten - An-schaffungs-kosten	Wie hoch sind die Kosten für die An-schaffung des Spracherkennungstools? Die Bewertung erfolgt folgendermaßen: 5 = Keine Kosten 4 = *Keine Wertung bei diesem Kriterium.* 3 = Moderate Kosten 2 = *Keine Wertung bei diesem Kriterium.* 1 = Hohe Kosten	Die Kosten bilden ein wichti-ges Kriterium. Die Gewich-tung der Anschaffungskosten entspricht der Gewichtung der laufenden Kosten. → Gewichtung: **4%**
Kosten – lau-fende Kosten	Wie hoch sind die laufenden Kosten für das Spracherkennungstool? Dies kann beispielsweise Lizenzkosten, die jährlich anfallen, beinhalten. Die Bewertung erfolgt folgendermaßen: 5 = Keine Kosten	Die Kosten bilden ein wichti-ges Kriterium. Die Gewich-tung der Anschaffungskosten entspricht der Gewichtung der laufenden Kosten. → Gewichtung: **4%**

	4 = *Keine Wertung bei diesem Kriterium.* 3 = Moderate Kosten 2 = *Keine Wertung bei diesem Kriterium.* 1 = Hohe Kosten	
Gesamtbewertung	Dies beinhaltet das zusammengefasste Ergebnis der Kriterien. Die einzelnen Bewertungen werden entsprechend der prozentualen Gewichtung aufsummiert. Je höher das Ergebnis der Gesamtbewertung ist, desto besser ist das Spracherkennungstool.	

Tabelle 3: Kriterienkatalog mit Beschreibung

3.3.2 Bewertung der beschriebenen Tools

Die in Kapital 3.1 und Kapital 3.2 beschriebenen Spracherkennungstools werden entsprechend des Kriterienkatalogs analysiert und bewertet. Somit wird ermittelt, ob das jeweilige Spracherkennungstool eine mögliche Lösung für die Applikation der _Versicherung ist.

3.3.2.1 Bewertung von CMU Sphinx

Im folgenden Abschnitt wird das Tool Sphinx 4 anhand des in Kapitel 3.1.1 dargestellten Kriterienkatalogs bewertet.

Kriterium	Gewichtung	Bewertung (inklusive Beschreibung)	Ergebnis
Einsatzgebiete	0%	Bedienung des Computers ohne Maus und Tastatur v.a. für körperbehinderte Personen und Senioren.	0
Funktionalitäten	4%	4 Es besteht die Möglichkeit ein Trainingstool mit zu installieren.	16

Vokabular – Anzahl der Sprachen	2%	2 Das Tool kann in Englisch oder Deutsch heruntergeladen werden, jedoch ist die deutsche Version nicht Aktuell.	4
Vokabular – Sprachen	7%	5 Die deutsche Version des Tools wurde nicht weiterentwickelt und es gibt keine Angaben, ob das Tool in der Sprache weiterentwickelt wird.	35
Vokabular – Wortschatz/ Themenbereiche	3%	3	9
Vokabular - Umfang	0%	Es wurden keine Angaben zum Umfang des Vokabulars gefunden.	0
Vokabular – Import	6%	1	6
Kompatibilität – Endgeräte	6%	5	30
Kompatibilität – Betriebssysteme	6%	5	30
Kompatibilität – Programmiersprachen	4%	2	8
Technische Anforderungen	0%	Es wurden keine Angaben zu den technischen Anforderungen gefunden.	0
Offline-Modus	5%	1	5
Installationsaufwand	6%	2	12
Registrierung	5%	5	25
Sprecherabhängigkeit	8%	1	8

		Das Tool muss trainiert werden und ist damit Sprecherabhängig.	
Sprachadaptives System	8%	1 Das Tool kann trainiert werden, lernt jedoch nicht automatisch mit der Nutzung dazu.	8
Dokumentation	8%	3 Es ist nur eine englische Version einer Dokumentation zu finden, die jedoch sehr umfangreich ist.	24
Datenschutz	8%	1	8
Lizenzen	6%	5	30
Kosten - Anschaffungskosten	4%	5	20
Kosten – laufende Kosten	4%	5	20
Gesamtbewertung	100%		**298** (max. 500)

Tabelle 4: Kriterienkatalog für CMU Sphinx

CMU Sphinx ist ein gut funktionierendes Spracherkennungstool, welches jedoch ausschließlich in der englischen Version genutzt werden kann. Der Entwicklungsstand des deutschsprachigen Tools ist nicht aktuell. Des Weiteren ist die Weiterentwicklung nicht gegeben. Daher eignet sich CMU Sphinx nicht für die Nutzung in der deutschen Sprache.

3.3.2.2 Bewertung von RWTH ASR

In der folgenden Tabelle wird das Tool RWTH ASR anhand der Kriterien bewertet.

Kriterium	Gewichtung	Bewertung (inklusive Beschreibung)	Ergebnis
Einsatzgebiete	0%	Bedienung des Computers ohne Maus und Tastatur, Diktieren von Texten, Grundlage (Framework) zur Nutzung von Spracherkennungssystemen	0
Funktionalitäten	4%	5 Das Tool bietet umfangreiche Funktionen, d.h. es lässt sich beliebig mit Vokabeln erweitern (Import und Export), kann Sprache erkennen, den PC steuern und sich an den Sprecher anpassen.	20
Vokabular – Anzahl der Sprachen	2%	0 Im Download der Software sind keine akustischen oder sprachlichen Modelle enthalten. Diese müssten dann ggf. von einer Datenbank bezogen werden.	0
Vokabular – Sprachen	7%	0 Da keine sprachlichen Modelle vorhanden sind, muss auch dieser Punkt negativ bewertet werden.	0
Vokabular – Wortschatz/ Themenbereiche	3%	0 Kein Vokabular vorhanden.	0

Vokabular - Umfang	0%	0 Kein Vokabular vorhanden. Nur im Demo-System sind 60.000 englische Vokabeln hinterlegt.	0
Vokabular – Import	6%	5 Vokabeln lassen sich leicht per XML oder anderen Text-formaten importieren.	30
Kompatibilität – Endgeräte	6%	1 Das System wurde für nor-male Desktop-PCs entwi-ckelt. Anpassung an andere Geräte könnte möglich sein.	6
Kompatibilität – Betriebssys-teme	6%	3 RWTH ASR wird aktuell für Linux (x86 und x86-64) und MAC OS X (Intel) unter-stützt.	18
Kompatibilität – Program-miersprachen	4%	1 Programmiert wurde in C++. Über die Kompatibilität zu anderen Sprachen erfolgt kein Hinweis.	4
Technische Anforderungen	0%	Über die technischen Anfor-derungen sind leider keine Informationen zu bekom-men.	0
Offline-Modus	5%	5 Das Tool wird ausschliess-lich offline genutzt.	25
Installationsaufwand	6%	1	6

		Der Installationsaufwand wird sehr hoch eingeschätzt und erfordert eine fachkundige Hand. Mehrere Bibliotheken und Tools werden vorausgesetzt.	
Registrierung	5%	3 Um das Softwarepaket herunterzuladen muss sich der Anwender einmalig registrieren und erhält dann den Downloadlink.	15
Sprecherabhängigkeit	8%	5 Das System passt sich mit der Zeit an den Sprecher an, ist aber nicht von ihm abhängig.	40
Sprachadaptives System	8%	5 Mit der Zeit passt sich das System an den Nutzer an. Unter anderem werden beliebte Redewendungen, Atempausen etc. berücksichtigt.	40
Dokumentation	8%	5 Eine umfangreiche Dokumentation sämtlicher Softwarekomponenten und theoretischer Überlegungen stehen zum Abruf bereit (Registrierung notwendig!).	40
Datenschutz	8%	1 Über den Datenschutz sind	8

		keine Informationen vorhanden.	
Lizenzen	6%	5 Es sind keine Lizenzen notwendig. Es muss lediglich der RWTH ASR License zugestimmt werden, die eine kostenlose Nutzung garantiert, inklusive der Weiterverbreitung und Modifikation der Software für nicht kommerzielle Zwecke.	30
Kosten - Anschaffungskosten	4%	5 Keine Anschaffungskosten notwendig – Open Source.	20
Kosten – laufende Kosten	4%	5 Keine laufenden Kosten fallen an, ausser für die eigene Weiterentwicklung.	20
Gesamtbewertung	100 %		**322** (max. 500)

Tabelle 5: Kriterienkatalog für RWTH ASR

RWTH ASR landet mit der Gesamtbewertung im Mittelfeld des Testfeldes. Grund dafür sind vor allem die fehlenden akustischen und sprachlichen Modelle, mit denen eine deutlich bessere Platzierung möglich gewesen wäre. Ansonsten bietet das Tool eine gute Grundlage zur Weiterentwicklung. Positive Rezensionen sollen außerdem aus Forschung- und Entwicklungsprogrammen kommen. Wermutstropfen allerdings die fehlende Unterstützung für die kommerzielle Nutzung.

3.3.2.3 Bewertung von Simon

In der folgenden Tabelle 6 wird das Spracherkennungstool Simon bewertet.

Kriterium	Gewichtung	Bewertung (inklusive Beschreibung)	Ergebnis
Einsatzgebiete	0%	Bedienung des Computers ohne Maus und Tastatur v.a. für körperbehinderte Personen und Senioren	0
Funktionalitäten	4%	5 Verschiedene Funktionalitäten – unter anderem: Import und Export von Vokabeln Import von Grammatik Training von Simon Programm starten Öffnung einer URL Verknüpfung eines Kommandos mit einem Text Steuerung der Computermaus Diktat Künstliche Intelligenz Taschenrechner Virtuelle, sprachgesteuerte Tastatur Sprachausgabe	20
Vokabular – Anzahl der Sprachen	2%	3	6
Vokabular – Sprachen	7%	5 Mögliche Sprachen: Deutsch Englisch Spanisch	35
Vokabular – Wortschatz/ Themenbereiche	3%	3	9

Vokabular - Umfang	0%	Keine Informationen vorhanden.	0
Vokabular – Import	6%	5 Es ist möglich, Wörter zu hinzuzufügen und zu entfernen.	30
Kompatibilität – Endgeräte	6%	1 Simon läuft auf dem normalen Computer.	6
Kompatibilität – Betriebssysteme	6%	5 Simon läuft auf den Betriebssystemen von Windows und Linux. Aktuell wird an einer Version für Mac OS gearbeitet. Für Entwickler ist der Source Code für Windows, Linux und Mac OS verfügbar.	30
Kompatibilität – Programmiersprachen	4%	5	20
Technische Anforderungen	0%	Keine speziellen technischen Anforderungen	0
Offline-Modus	5%	1 Aufgrund der Server-Client-Architektur ist eine Internetverbindung notwendig.	5
Installationsaufwand	6%	4	24
Registrierung	5%	3 Registrierung nicht für Simon selbst aber bspw. für HTK notwendig.	15
Sprecherabhängigkeit	8%	5	40
Sprachadaptives System	8%	1	8
Dokumentation	8%	5	40

Datenschutz	8%	1 Es liegen keine Informationen über Datenschutzmaßnahmen vor.	8
Lizenzen	6%	5	30
Kosten - Anschaf-fungskosten	4%	5 Simon ist ein Open Source-Programm.	20
Kosten – laufende Kosten	4%	5 Simon ist ein Open Source-Programm.	20
Gesamtbewertung	100%		**366** (max. 500)

Tabelle 6: Kriterienkatalog für Simon

Simon ist ein Spracherkennungstool, das viele Funktionalitäten bieten. Zudem kann es durch die flexiblen Implementierungsmöglichkeiten vielfältig verwendet werden. Da es zusätzlich über die deutsche Sprache verfügt, wird es als Lösung für die _Versicherung Applikation in Betracht gezogen.

3.3.2.4 Bewertung von Dragon NaturallySpeaking

Als letztes wird das kommerzielle Tool Dragon NaturallySpeaking (Version 13) bewertet.

Kriterium	Gewichtung	Bewertung (inklusive Beschrei-bung)	Ergebnis
Einsatzgebiete	0%	Nuance bietet verschiedene Versio-nen Ihrer Spracherkennungssoft-ware: Privat – Geschäftlich, mobil – Desktopanwendung, sowie für ver-schiedene Geschäftsbereiche (z.B. Versicherungswesen, Bauwesen etc.).	0
Funktionalitäten	4%	5	20

		Dragon NaturallySpeaking deckt zurzeit jede erdenkliche Funktionalität in Zusammenhang mit Spracherkennung ab.	
Vokabular – Anzahl der Sprachen	2%	5 Dragon ist in zehn Sprachen erhältlich und reicht vom amerikanischen Englisch über indisches Englisch bis hin zu Spanisch.	10
Vokabular – Sprachen	7%	5 Von den bewertbaren Sprachen sind alle bis auf Russisch, Chinesisch und Portugiesisch alle gängigen Sprachen vorhanden.	35
Vokabular – Wortschatz/ Themenbereiche	3%	5 Die Software ist für verschiedene Branchen (Versicherung, Rechtswesen, Telekommunikation usw.) zu haben. Demnach steht auch spezifisches Fachvokabular zur Verfügung, welches sich beliebig erweitern lässt.	15
Vokabular - Umfang	0%	Ab Kauf stehen ca. 150.000 Wortformen bereit, die sich mit Fachvokabular und individuellen Wörtern erweitern lassen.	0
Vokabular – Import	6%	5 Das Vokabular lässt sich über verschiedene Wege importieren. Beispielsweise können Dokumente hinzugefügt werden, aus denen dann unbekannte Wörter gefiltert werden. Eine andere Möglichkeit wäre die dauerhafte Verknüpfung mit Email-Programmen oder Kontaktbüchern.	30

Kompatibilität – Endgeräte	6%	5 Für alle Endgeräte gibt es eine Version, wobei die Desktopanwendung am umfangreichsten gestaltet ist. Mobile Versionen beschränken sich meist auf das einfache Diktieren von Mails oder SMS.	30
Kompatibilität – Betriebssysteme	6%	4 Dragon läuft auf mobilen Endgeräten mit iOs und Android. Auf Desktop-PCs muss MAC OS X, mind. Windows 7 oder Windows Server 2012 installiert sein. Für Linux gibt es keine Anwendung.	24
Kompatibilität – Programmiersprachen	4%	1 Leider sind keine Angaben über die Programmiersprache zu finden. Aber mit mind. einer Programmiersprache muss das Tool kompatibel sein.	4
Technische Anforderungen	0%	Die hier dargestellten Anforderungen beziehen sich auf die professionelle Lösung von Dragon: - mind. 2 GB RAM für 32-Bit-Versionen und 4 GB für 64-Bit-Versionen - Dual-Core Prozessor mit mind. 2,2 GHz Taktfrequenz und 2MB L2 Cache - Zur Installation sind 12 GB freier Speicher erforderlich, nach der Installation werden 8GB temporäre Daten wieder gelöscht - Soundkarte für 16-Bit Aufzeichnung - DVD-Laufwerk	0

		- Internetverbindung zur Produktaktivierung	
Offline-Modus	5%	5 Eine Offline-Verwendung ist möglich. Nur zur Aktivierung muss eine Internetverbindung bestehen.	25
Installationsaufwand	6%	5 Die Installation soll ca. 20min in Anspruch nehmen. Für die Datenmenge eine ordentliche Zeit. Danach wird der Nutzer durch das Programm geführt.	30
Registrierung	5%	5 Das Programm muss online aktiviert werden (anonym), erfordert aber keine Registrierung.	25
Sprecherabhängigkeit	8%	1 Es wird für jeden Benutzer ein Profil erstellt, an das sich das Tool anpasst, die Gewohnheiten erkennt und übernimmt. Es ist somit abhängig von Sprecher.	8
Sprachadaptives System	8%	5 Das System lernt mit der Zeit dazu, und passt sich an Sprachgewohnheiten des Nutzers an. So können Sätze einfacher identifiziert und umgewandelt werden.	40
Dokumentation	8%	5 Für Dragon NaturallySpeaking steht eine umfangreiche Dokumentation zur Bedienung parat. Des Weiteren hat der Anwender Zugriff auf den	40

		Support von Nuance Communications, sodass alle Fragen beantwortet werden können.	
Datenschutz	8%	5 Die Aktivierung der Software erfolgt anonym und über eine verschlüsselte Verbindung.	40
Lizenzen	6%	3 Jede Kopie von NaturallySpeaking enthält nur eine Lizenz für einen Nutzer. Für Unternehmenslösungen lässt sich das Lizenzvolumen individuell verhandeln.	18
Kosten - Anschaffungskosten	4%	1 Es gibt keine Informationen über Kosten für die geschäftliche Nutzung. Die private Version steigt jedoch erst bei 99 Euro ein – pro Lizenz!	4
Kosten – laufende Kosten	4%	5 Angesichts der hohen einmaligen Anschaffungskosten sollten keine laufenden Kosten anfallen.	20
Gesamtbewertung	100 %		**418** (max. 500)

Tabelle 7: Kriterienkatalog für Dragon NaturallySpeaking

Dragon NaturallySpeaking ist bis auf ein paar kleine Ausnahmen durchweg sehr gut zu bewerten und berechtigt die Stellung als Marktführer unter den kommerziellen Tools. Viele zukünftige Open Source-Tools werden sich daran orientieren.

3.3.3 Vergleich der Bewertungsergebnisse

Die folgende Tabelle 8 stellt die Gesamtbewertung der geprüften Tools dar:

	Dragon NaturallySpeaking	CMU Sphinx	RWTH ASR	Simon
Bewertung	**418**	**298**	**322**	**366**

Tabelle 8: Übersicht Bewertungsergebnisse

Bei der Betrachtung der Ergebnisse der Open Source-Tools CMU Sphinx, RWTH ASR und Simon ist Simon das beste Tool, da es die höchste Gesamtpunktzahl hat. Zudem ist Simon vorteilhaft, da es in deutscher Sprache verfügbar ist. Durch den vorhandenen Source Code kann es flexibel eingesetzt werden.

Beim Vergleich des kommerziellen Spracherkennungstools Dragon NaturallySpeaking mit den Open Source-Tools ist ein deutlicher Abstand erkennbar. Dragon NaturallySpeaking ist mit einer Gesamtpunktzahl von 418 wesentlich besser als die Open Source-Tools. Eine mögliche Begründung für den Qualitätsunterschied ist, dass die Open Source-Tools zum Teil veraltet sind oder momentan noch entwickelt werden.

3.4 Praxistest der Spracherkennungstools

Nach der Bewertung mit Hilfe des Kriterienkatalogs wird in den folgenden Kapiteln ein Test durchgeführt, um praktische Bewertungen miteinzubeziehen.

3.4.1 Auswahl der zu testenden Spracherkennungstools

Für den praktischen Test der Spracherkennungstools sind generell die folgenden vier Tools möglich:

- CMU Sphinx,
- RWTH ASR,
- Simon,
- Dragon NaturallySpeaking.

Bei CMU Sphinx wurden die gesprochenen Wörter nicht erkannt. Grund hierfür ist möglicherweise, dass die verwendeten Sprachmodelle und das Tool selbst noch nicht reif genug sind. Es sind somit keine Aussagen über die Qualität von CMU Sphinx möglich.

Bei RWTH ASR ist nur eine Demo-Version für das Betriebssystem Linux vorhanden. Wir haben schließlich einen Kommilitonen gefunden, der einen Computer mit Linux-Betriebssystem hat. Dieser hat versucht, die Demo-Version von RWTH ASR zu installieren. Die heruntergeladene Datei konnte allerdings nicht ausgeführt werden. Deshalb konnte RWTH ASR nicht getestet werden.

Das Spracherkennungstool Simon steht als Version für Windows-Betriebssysteme zur Verfügung. Allerdings wird das Diktierplugin für Simon noch entwickelt. Somit war ein Test von Simon ebenfalls nicht möglich.

Da Dragon NaturallySpeaking ein kommerzielles Spracherkennungstool ist, ist nur eine kostenpflichtige Version des Tools verfügbar. Allerdings gibt es eine kostenlose App namens Dragon Dictation. Dragon Dictation ist eine App für die iOS-Systeme von Apple. Die App läuft auf dem iPhone und dem iPad. Schließlich wurde Dragon Dictation auf dem iPad für den praktischen Test verwendet.

3.4.2 Testkriterien

Für den Test der Spracherkennungstools wurden Kriterien festgelegt, anhand deren die ausgewählten Spracherkennungstools getestet werden soll. Die Festlegung der Kriterien orientiert sich vor allem daran, was durch einen praktischen Test bewertet werden kann, wie zum Beispiel die Erkennungsgenauigkeit. Nach der Identifikation der Kriterien wurden die Bewertungen von eins bis fünf und die jeweilige Gewichtung wie beim Kriterienkatalog ermittelt.

Die Testkriterien werden in der folgenden Tabelle 9 beschrieben:

Kriterium	Beschreibung	Gewichtung (mit Begründung)
Erkennungsgenauigkeit – Wortkorrektheit W_K	Es wird gemessen, wie zutreffend das Spracherkennungstool das Gesprochene in geschriebenen Text umwandelt. Bei diesem Kriterium wird die Wortkorrektheit geprüft. Diese betrachtet, inwiefern Worte verwechselt (N_V) oder ausgelassen (N_A) werden. Für die Berechnung von W_K wird folgende Formel angewandt: $$W_K = \frac{N_W - N_V - N_A}{N_W}$$ N_W: Anzahl der gesprochenen Wörtern Es wird der folgende Satz getestet: „Heute präsentieren wir unsere Projektarbeit, die das Thema Spracherkennung behandelt.“ Die Ergebnisse von W_K werden bewertet, wie folgt: 5 = > 0,9 4 = 0,6-0,9 3 = 0,5 2 = 0,1-0,4 1 = < 0,1	Die Erkennungsgenauigkeit ist äußerst wichtig, da das Spracherkennungstool das Gesprochene korrekt erkennen und umwandeln muss. Nur wenn eine korrekte Umwandlung stattfindet, ist das Tool effektiv und kann für die Spracherkennung eingesetzt werden. Die Wortkorrektheit und die nachfolgende Wortakkuratheit haben die gleiche Gewichtung und bilden zusammen die Bewertung für die Erkennungsgenauigkeit. → Gewichtung: **7,4%**
Erkennungsgenauigkeit – Wortakkuratheit W_A	Es wird gemessen, wie zutreffend das Spracherkennungstool das Gesprochene in geschriebenen Text umwandelt. Bei diesem Kriterium wird die Wortakkuratheit geprüft. Diese betrachtet nicht nur, ob Worte verwechselt (N_V) oder ausgelassen (N_A) werden, sondern auch, inwiefern falsche Worte eingefügt (N_E) wer-	Die Erkennungsgenauigkeit ist äußerst wichtig, da das Spracherkennungstool das Gesprochene korrekt erkennen und umwandeln muss. Nur wenn eine korrekte Umwandlung stattfindet, ist das Tool effektiv und kann für die Spracherkennung eingesetzt

	den. Für die Berechnung von W_A wird folgende Formel angewandt: $W_K = \frac{N_W - N_V - N_A - N_E}{N_W}$ N_W: Anzahl der gesprochenen Wörter Es wird der folgende Satz getestet: „Heute präsentieren wir unsere Projektarbeit, die das Thema Spracherkennung behandelt.“ Die Ergebnisse von W_K werden bewertet, wie folgt: 5 = > 0,9 4 = 0,6-0,9 3 = 0,5 2 = 0,1-0,4 1 = < 0,1	werden. Die Wortakkuratheit und vorherige Wortkorrektheit haben die gleiche Gewichtung und bilden zusammen die Bewertung für die Erkennungsgenauigkeit. → Gewichtung: **7,4%**
Fachvokabular	Die Erkennungsgenauigkeit von Fremdwörtern und fachspezifischen Begriffe wird gemessen. Es werden folgende Fachbegriffe getestet: - „Desoxyribonukleinsäure“ - „Photovoltaikanlagen-Versicherung“ - „Schadenminderungspflicht“ -„Karyogramm“ Die Bewertung erfolgt entsprechend der Anzahl an erkannten Fachbegriffen: 5 = alle 4 erkannten Worte 4 = 3 erkannte Worte 3 = 2 erkannte Worte	Das Vorhandensein von Fachvokabular hat keine hohe Gewichtung für den Test des Spracherkennungstools. Falls Fachvokabular vorhanden ist, ist das nur ein zusätzliches Feature, das allerdings nicht unbedingt notwendig ist. Gegebenenfalls ist es möglich Fachbegriffe zum Vokabular des Tools hinzuzufügen. → Gewichtung: **7,4%**

	2 = 1 erkanntes Wort 1 = keine erkannten Worte	
Dialekte	Dieses Kriterium beschreibt, inwiefern die Spracherkennung trotz verschiedener Dialekte, wie Bayrisch, Schwäbisch oder Niederdeutsch, funktioniert. Es wird der folgende Satz in einigen Dialekten getestet: „Ich bin heute Morgen aufgewacht, habe das Licht angemacht und ein wenig in meiner Zeitung gelesen, dann habe ich mir ein Brot mit Marmelade geschmiert.“ Die Bewertung erfolgt folgendermaßen: 5 = Dialektische Begriffe (Bsp.: Gsälz) und sehr starke dialektische Aussprache werden erkannt. 4 = *Keine Wertung bei diesem Kriterium.* 3 = Leicht dialektische Aussprache wird erkannt. 2 = *Keine Wertung bei diesem Kriterium.* 1 = Keine dialektische Aussprache wird erkannt.	Die Erkennung von Dialekten hat keine hohe Gewichtung. Es ist wünschenswert, wenn leicht dialektische Aussprache erkannt wird. Es wird davon ausgegangen, dass die Nutzer mit leichten Abweichungen Hochdeutsch sprechen, wenn sie das Tool nutzen. → Gewichtung: **9,25%**
Sprachtoleranz – Füllwörter	Es wird überprüft, inwiefern Füllwörter, wie zum Beispiel „äh“ oder „ähm“, von System erkannt und bei der Umwandlung ausgeblendet werden. Es wird folgender Satz getestet: „Heute ähm präsentieren wir unsere Projektarbeit, die das Thema äh Spracherkennung behandelt.“ Die Bewertung erfolgt folgendermaßen: 5 = Alle Füllwörter werden erkannt und ausgeblendet.	Die Erkennung und Ausblendung von Füllwörtern hat eine relativ geringe Gewichtung. Für die Erstellung von Berichten ist es wünschenswert, dass Füllwörter erkannt und ausgeblendet werden. Ist dies nicht der Fall, entsteht ein hoher Aufwand für die Nachbearbeitung der umgewandelten Berichte, da alle Füllworte manuell entfernt werden müssen.

	4 = *Keine Wertung bei diesem Kriterium.* 3 = Die Füllwörter werden erkannt und mit ausgegeben. 2 = *Keine Wertung bei diesem Kriterium.* 1 = Füllwörter werden nicht erkannt und falsche Ergebnisse werden geliefert.	Füllwörter dürfen keinesfalls dazu führen, dass falsche Worte erkannt werden. Das würde dazu führen, dass das Spracherkennungstool ineffizient ist, da das falsche Ergebnis zurückgeliefert wird. → Gewichtung: **7,4%**
Sprachtoleranz - Sprachstörungen	Sprachstörungen umfassen beispielsweise Stottern oder unklare Aussprachen, wie Nuscheln. Es wird der folgende Satz getestet: „Als erstes wird Stottern und als zweites Nuscheln getestet.“ Die Bewertung erfolgt folgendermaßen: 5 = Stottern und Nuscheln haben keine Auswirkung auf das Ergebnis. 4 = Nuscheln verfälscht das Ergebnis. 3 = Stottern verfälscht das Ergebnis 2 = Stottern und Nuscheln verfälschen das Ergebnis, aber nur leichte Abweichungen. 1 = Stottern und Nuscheln verfälschen das Ergebnis, aber größere Abweichungen.	Die Gewichtung für Sprachstörungen ist relativ hoch. Sprachstörungen sollten das Ergebnis der Spracherkennung nicht verfälschen. Die Bewertung ist unterschiedlich entsprechend verschiedener Sprachstörungen: **Stottern**: Stottern soll keinen Einfluss auf die Spracherkennung haben. Falls ein Nutzer stottert, kann das nicht einfach von der Person unterdrückt werden. **Nuscheln**: Im Gegensatz zum Stottern ist der Einfluss von Nuscheln auf die Spracherkennung nicht so wichtig. Die Nutzer können versuchen, eine deutliche Aussprache anzuwenden. Somit kann die Spracheingabe ohne Nuscheln durchgeführt werden. → Gewichtung: **11,1%**
Geschwindigkeit der Sprach-	Wie schnell liefert das Tool den umgewandelten Text zu. Inwiefern das Ergebnis korrekt ist, wird bei diesem Kriterium nicht betrachtet. Die Korrektheit wird bei dem Kriterium „Erkennungsgenauigkeit“	Es ist wichtig, dass die Spracherkennung relativ schnell funktioniert. Beispielsweise wenn es mehrere Minuten dauert, bis ein Satz umgewandelt wird, ist die

erken-nung	geprüft. Es werden die folgenden Absätze geprüft: „Dragon Dictation ist eine Spracherkennungssoftware auf der Basis von Dragon NaturallySpeaking®, mit der Sie einfach per Sprache SMS oder E-Mails diktieren. Der Text erscheint innerhalb von Sekunden auf Ihrem iPhone und kann sofort versendet werden. Diese Methode ist bis zu fünf Mal schneller als die Eingabe über Tastatur. Mit Dragon Dictation können Sie auch Ihren Facebook-Status aktualisieren, sich selbst Notizen oder Erinnerungen senden oder Tweets für die ganze Welt veröffentlichen … alles per Sprache. Wenn Sie also unterwegs sind, setzen Sie lieber Ihre Stimme statt der Tastatur ein, um kurze SMS oder längere E-Mails zu verfassen.“[106] Es werden Wertungen entsprechend der folgenden Liste vorgenommen: 5 = 0-2 Sekunden 4 = 3-5 Sekunden 3 = 6-10 Sekunden 2 = 11-15 Sekunden 1 = ab 15 Sekunden	Umwandlung äußerst ineffizient und das Tool ist für den Einsatzzweck nicht geeignet. → Gewichtung: **14,8%**
Ge-schwin-digkeit der Sprach-erken-	Das Kriterium überprüft, inwiefern die mündliche Spracheingabe inklusive Umwandlung in Text schneller ist als die Tastatureingabe. Der Testsatz wird aus der vorherigen	Der Vergleich mit der Tastatureingabe ist sehr wichtig. Falls die Tastatureingabe schneller ist als das Spracherkennungstool, besteht keine Notwendigkeit für die Einführung des Spracherken-

[106] Apple Inc. (2014)

nung – Vergleich mit Tastatureingabe	Kategorie entnommen. Die Bewertung erfolgt folgendermaßen: 5 = Spracherkennung ist schneller. 4 = *Keine Wertung bei diesem Kriterium.* 3 = Sind gleich schnell. 2 = *Keine Wertung bei diesem Kriterium.* 1 = Tastatureingabe ist schneller.	nungstools. → Gewichtung: **14,8%**
Bedienbarkeit	Es wird geprüft, wie gut bzw. einfach die Bedienung des Spracherkennungstools ist. Dies beinhaltet, ob das Einlernen des Nutzers notwendig ist oder ob das Tool intuitiv bedient werden kann. Zudem wird die Anzahl an Arbeitsschritten betrachtet. Die Bewertung erfolgt folgendermaßen: 5 = Intuitive Bedienung ausreichend. 4 = *Keine Wertung bei diesem Kriterium.* 3 = Komplexe Bedienung, aber Anleitung vorhanden. 2 = *Keine Wertung bei diesem Kriterium.* 1 = Komplexe Bedienung und keine Anleitungen.	Die Bedienbarkeit ist ein wichtiges Kriterium. Eine komplexe Bedienung kann zur Ineffizienz des Spracherkennungstools führen, da eine Einarbeitung notwendig ist und die Nutzung des Spracherkennungstools länger dauert. Sind zusätzlich zur komplexen Bedienung keine Anleitungen vorhanden, ist der Aufwand für die Einarbeitung in das Spracherkennungstool äußerst hoch, da die Bedienung durch Tests ermittelt werden muss. → Gewichtung: **11,1%**
Hintergrundgeräusche	Es wird geprüft, inwiefern die Sprachumwandlung trotz Hintergrundgeräusche korrekt funktioniert. Dies kann wichtig sein, wenn das Tool in einem Umfeld mit hohem Geräuschpegel eingesetzt werden soll. Es wird folgender Satz getestet: „Heute präsentieren wir unsere Projektarbeit, die das Thema Spracherkennung behandelt." Die Bewertung erfolgt folgendermaßen:	Die effektive Spracherkennung bei Hintergrundgeräusche ist relativ wichtig. Falls das Spracherkennungstool in einem Umfeld mit hohem Geräuschpegel eingesetzt wird, sollte es trotzdem funktionieren. → Gewichtung: **9,25%**

	5 = Spracherkennung ohne Fehler 4 = Spracherkennung erfolgt mit geringen Fehlern (1 Fehler) 3 = Fehlermeldung erscheint, dass es zu viele Hintergrundgeräusche gibt. 2 = Spracherkennung erfolgt mit ein paar Fehlern (2-3 Fehlern) 1 = Viele Fehler bei der Spracherken-nung (ab 4 Fehlern)	
Tester-gebnis	Dies beinhaltet das zusammengefasste Ergebnis des Tests. Die einzelnen Be-wertungen werden entsprechend der prozentualen Gewichtung aufsummiert. Je höher das gesamte Testergebnis ist, desto besser ist das Spracherkennungs-tool.	

Tabelle 9: Testkriterien mit Beschreibung

3.4.3 Test von Dragon Dictation

Dragon Dictation ist eine mobile Spracherkennungsanwendung für Apples iOS-System (4.0 oder höher) und ist kompatibel mit iPhone, iPod Touch und iPad. Die Gratis-App bietet eine automatische Sprachumwandlung in Text und basiert auf der Desktopanwendung Dragon NaturallySpeaking von Nuance Communications. Seit Ende 2009 steht sie im Apple AppStore zur Verfügung – aktuell in der Version 2.0.28, die in der deutschen Version Gegenstand des Tests sein wird.[107]

Mit der App lassen sich schnell und einfach Texte diktieren, die dann als Email, Notiz oder Post in soziale Netzwerke genutzt werden. Voraussetzung ist jedoch eine Internetverbindung und eine maximale Länge der Spracheingabe von 60 Sekunden. Nachdem der User seinen Text in das interne Mikrofon des Geräts gesprochen hat, wird das Audiofile über die Dragon Server umgewandelt. Daraus ergibt sich der Vorteil, dass die Anwendung relativ klein gehalten wird (ca. 10MB). Sobald die Umwandlung vollzogen ist erscheint der Text auf dem Bildschirm des Geräts. Dies soll bis zu fünf Mal schneller als herkömmliche Tastatureingaben

[107] Vgl. Apple Inc. (2014)

geschehen. Sollten dann noch Fehler im Text auftauchen kann der Anwender die Korrekturfunktion mit den unterschiedlichen Vorschlägen nutzen oder individuelle Verbesserungen vornehmen.[108]

Dragon Dictation wird in diesem Test verwendet, da es auf dem kommerziellen Tool Dragon NaturallySpeaking basiert und gratis zur Verfügung steht.

Kriterium	Gewichtung	Bewertung (inklusive Beschreibung)	Ergebnis
Erkennungs-genauigkeit – Wortkorrektheit W_K	7,4%	5 Das Tool wertet den gesprochenen Satz ohne jegliche Fehler aus (vgl. Rechnung). $WK = \frac{10-0-0}{10} = 1$	37
Erkennungs-genauigkeit – Wortakkuratheit W_A	7,4%	5 Auch hier arbeitet das Tool fehlerfrei und fügt keine falschen Wörter ein (vgl. Rechnung). $WK = \frac{10-0-0-0}{10} = 1$	37
Fachvokabular	7,4%	4 Ein Wort wird leider nicht korrekt übernommen. Aus Karyogramm wird „Carrillo Gramm". Ansonsten werden die restlichen Wörter auf Anhieb erkannt.	29,6
Dialekte	9,25%	3 Der Satz wurde sowohl Berliner Dialekt, als auch im schwäbischen Dialekt getestet. Gängige Wörter, wie „Ick" und „I" statt „Ich" und „uffjewacht" für „aufgewacht", werden ohne Probleme erkannt. Besondere Begriffe müssen dem Tool beigebracht werden, sonst wird aus Gsälz ziemlich schnell Salz.	27,75

[108] Vgl. Apple Inc. (2014)

Sprachtoleranz – Füllwörter	7,4%	3 Viele Füllwörter wie „äh“ und „ähm“ werden sofort herausgefiltert. Andere, wie „also“ etc. müssen hingegeben manuell gefiltert werden.	22,2
Sprachtoleranz - Sprachstörungen	11,1%	1 Leider tut sich das Programm sehr schwer mit Sprachstörungen. Viele Wörter des leichten Testsatzes wurden gar nicht erkannt. Vielleicht passt sich das Programm mit längerer Nutzungsdauer und Korrektur an den Nutzer besser an.	11,1
Geschwindigkeit der Spracherkennung	14,8%	4 Die gesprochenen Absätze wurden sehr schnell (ca. 3 Sekunden) in Text umgewandelt. Generell gilt jedoch, dass die Länge des gesprochenen Textes und die Verwendung von Fachvokabular die Umwandlungsdauer beeinflussen.	74
Geschwindigkeit der Spracherkennung – Vergleich mit Tastatureingabe	14,8%	5 Mit der Spracherkennung (Spracheingabe + Umwandlung) wurde der Absatz in 43 Sekunden in das vorhandene Formular eingetragen. Die Tastatureingabe dauerte mit 2:09 Minuten fast drei Mal so lange. (Hinweis: Kein 10-Finger-System, aber relativ schnelle Schreibweise). Besonders bei noch längeren Texten wird die Spracherkennungssoftware ihre Geschwindigkeit ausspielen können.	74
Bedienbarkeit	11,1%	5 Die Bedienung wurde sehr einfach gehalten, ist aber völlig ausreichend für die Anwendung. Viele Schritte erklären sich	55,5

		von selbst. Für häufig verwendete Befehle bzw. Satzzeichen gibt es eine Anleitung zum Nachlesen.	
Hintergrundge-räusche	9,25%	2 Während des Tests lief im Hintergrund Musik, um die Arbeit in einem Groß-raumbüro mit vielen Kollegen zu simulie-ren. Leider schlichen sich immer wieder Fehler ein, in dem Wörter nicht eingefügt oder seltsame Wortgebilde geschaffen wurden.	18,5
Testergebnis	100 %		**384,45** (max. 500)

Tabelle 10: Kriterienkatalog der Auswertung von Dragon Dictation

Dragon Dictation schneidet in diesem Test gut ab. In ruhiger Umgebung konnte es seine Stärken in Erkennungsgenauigkeit, Bedienbarkeit und Geschwindigkeit ausspielen. Sobald die Sprache nicht eindeutig zu erkennen war schlichen sich mehrere Fehler ein. Möglicherweise legen sich einige Fehler über einen längeren Testzeitraum. Großer Vorteil der App: Sie ist kostenlos.

4 Zusammenfassung und Ausblick

Diese Projektarbeit bestand aus zwei Aufgaben. Zum einen wurden die allgemeine und die technische Funktionsweise von Spracherkennung erläutert. Anschließend wurde eine Marktanalyse durchgeführt. Es wurde ermittelt, welche kommerziellen und Open Source Spracherkennungstools auf dem Markt zu finden sind. Anschließend wurde analysiert, welche dieser Tools sich als mögliche Lösung für die _Versicherung Applikation zur Schadensaufnahmen eignen.

Zunächst wurde recherchiert, welche Spracherkennungstools aktuell auf dem Open Source und kommerziellen Markt vorhanden sind. Als Open Source-Tools wurden CMU Sphinx, Julius, RWTH ASR, Simon und VoxForge identifiziert. Da Nuance Dragon der Marktführer bezüglich Spracherkennungstools ist, wurde als kommerzielles Tool Dragon NaturallySpeaking ausgewählt.

Nach der Auswahl möglicher Tools, wurden anhand eines Kriterienkatalogs sowohl die Open Source-Tools als auch das kommerzielle Tool bewertet. Daraus ergab sich, dass Simon das beste Open Source-Tool ist. Allerdings weist das kommerzielle Tool Dragon NaturallySpeaking wesentlich höhere Qualität auf. Eine mögliche Begründung hierfür ist, dass die Open Source-Tools teilweise veraltet sind oder aktuell entwickelt werden.

Im Anschluss an die Bewertungen mit Hilfe des Kriterienkatalogs wurde das kommerzielle Tool in der Praxis getestet. Die Open Source-Tools konnten nicht getestet werden, da sie entweder veraltet oder noch in der Entwicklung sind. Aufgrund der relativ hohen Kosten für Dragon NaturallySpeaking wurde die kostenfreie App namens Dragon Dictation für Apple iOS getestet. Durch den praktischen Test wurde bestätigt, dass die Diktierfunktion von Dragon sehr gut ist und sich somit als kommerzielles Tool für die _Versicherung Applikation eignet.

Es liegt schließlich im Ermessen der _Versicherung, ob eine Open Source oder eine kommerzielle Lösung erwünscht ist. Da die Open Source-Tools veraltet bzw. in der Entwicklung sind, entsteht ein großer Aufwand für die Entwicklung und Einführung einer Open Source-Lösung. Im Gegensatz dazu sind kommerzielle Tools mit hohen Anschaffungskosten verbunden. Sie können aber einfacher und schneller implementiert werden.

Falls eine Open Source-Lösung gewählt wird, ist das Spracherkennungstool Simon zu empfehlen. Allerdings muss für die Verwendung von Simon eine Applikation entwickelt werden. Dies kann Aufgabe in einem nachfolgenden Projekt sein.

Quellenverzeichnisse

Literaturverzeichnis

Euler, S. (2006): Grundkurs Spracherkennung, Vom Sprachsignal zum Dialog – Grundlagen und Anwendung verstehen – Mit praktischen Übungen, 1. Auflage, Wiesbaden: Friedr.Vieweg & Sohn Verlag

Glaser, E. u. a. (2014): Sprache(n) verstehen, Zürich: vdf Hochschulverlag

Krapichler, C. (1999): Eine neue Mensch-Maschine-Schnittstelle für die Analyse medizinischer 3D-Bilddaten in einer virtuellen Umgebung, München: Herbert Utz Verlag

Müller, J. (1997): Die semantische Gliederung zur Repräsentation des Bedeutungsinhalts innerhalb sprachverstehender Systeme, München: Herbert Utz Verlag

Oswald, P./Goebl, H./Brix, E. (2001): Der Mensch und seine Sprache(n), Wien: Böhlau Verlag

Pfister, B./Kaufmann, T. (2008): Sprachverarbeitung: Grundlagen und Methoden der Sprachsynthese und Spracherkennung, Berlin/Heidelberg: Springer-Verlag

Schukat-Talamazzini, E.G. (1995): Automatische Spracherkennung, Statistische Verfahren der Musteranalyse, ohne Auflage, ohne Ort: Vieweg Verlag

Stadermann, Dipl.-Ing.J. (2005): Automatische Spracherkennung mit hybriden akustischen Modellen, Lehrstuhl für Mensch-Maschine Kommunikation Technische Universität München: München

Wendemuth A. (2004): Grundlagen der stochastischen Sprachverarbeitung, Oldenbourg Wissenschaftsverlag GmbH: München

Verzeichnis der Internet- und Intranet-Quellen

Apple Inc. (2014): Dragon Dictation, https://itunes.apple.com/de/app/dragon-dictation/id341446764?mt=8, Abruf: 13.01.2015

Beer D. (o. J.): Spracherkennung mit Hidden-Markov-Modells (HMMs), http://www-mmdb.iai.uni-bonn.de/lehre/proprak0304/beer.pdf, Abruf: 18.12.2014

Berger, P. (2005): Seminar Digitale Signalverarbeitung, http://userpages.uni-koblenz.de/~physik/informatik/DSV/FFT.pdf, Abruf: 17.01.2015

Bibliographisches Institut GmbH (2013a):

Phonem, Fonem, das, http://www.duden.de/node/701348/revisions/1259166/view, Abruf: 25.12.2014

Bibliographisches Institut GmbH (2013b):

Spracherkennung, http://www.duden.de/node/681999/revisions/1095212/view, Abruf: 01.12.2014

Bibliographisches Institut GmbH (2013c):

Spracherkennungsprogramm, http://www.duden.de/node/708772/revisions/1263112/view, Abruf: 01.12.2014

Bibliographisches Institut GmbH (2013d):

transkribieren, http://www.duden.de/node/761326/revisions/1292591/view, Abruf: 17.01.2015

CHIP Digital GmbH (2013):

Die Siri-Story: Vom Militärprojekt zum Apple-Dienst, http://www.chip.de/news/Die-Siri-Story-Vom-Militaerprojekt-zum-Apple-Dienst_60064765.html, Abruf: 25.11.2014

CMU Sphinx (o. J.a): CMU Sphinx Downloads, http://cmusphinx.sourceforge.net/wiki/download/, Abruf: 22.12.2014

CMU Sphinx (o. J.b): CMU Sphinx, http://cmusphinx.sourceforge.net/, Abruf: 22.12.2014

CMU Sphinx (o. J.c): Training Acoustic Model For CMUSphinx, http://cmusphinx.sourceforge.net/wiki/tutorialam, Abruf: 22.12.2014

CMU Sphinx (o. J.d): Generating a dictionary, http://cmusphinx.sourceforge.net/wiki/tutorialdict, Abruf: 22.12.2014

Dernbach, C. (2012): Audrey, IBM Shoebox, ViaVoice, Dragon, Siri: Die Geschichte der automatischen Spracherkennung, http://www.mr-gadget.de/tech-history/2012-04-20/audrey-ibm-shoebox-viavoice-dragon-siri-die-geschichte-der-automatischen-spracherkennung, Abruf: 25.11.2014

Eichelsbacher, P. (o. J.): Stochastische Algorithmen I., http://www.ruhr-uni-bo-chum.de/imperia/md/content/stochastik/veroeffentlichungen/quicksort.pdf, Abruf: 27.12.2014

gbt.ch (o. J.): Kriterienkatalog, http://www.gbt.ch/Lexikon/K/Kriterienkatalog.html, Abruf: 20.01.2015

Grasch, P. (2014): Simon on OS X, http://grasch.net/node/29, Abruf: 13.01.2015

Grasch, P. (o. J.a): About Simon, https://simon.kde.org, Abruf: 13.01.2015

Grasch, P. (o. J.b): The Simon Handbook, https://docs.kde.org/development/en/extragear-accessibility/simon/simon.pdf, Abruf: 15.01.2015

Herbert Utz Verlag GmbH (o. J.): Johannes Müller: Die semantische Gliederung zur Repräsentation des Bedeutungsinhalts innerhalb sprachverstehender Systeme, http://www.utzverlag.de/shop.php?bn=31210, Abruf: 27.12.2014

Julius development team (2014a): Open-Source Large Vocabulary CSR Engine Julius, http://julius.sourceforge.jp/en_index.php?q=index-en.html#feature, Abruf: 18.12.2014

Julius development team (2014b): Open-Source Large Vocabulary CSR Engine Julius, http://julius.sourceforge.jp/en_index.php, Abruf: 15.01.2015

KDE e.V. (2014): Simon/Installation, https://userbase.kde.org/Simon/Installation, Abruf: 13.01.2015

Konradin Medien GmbH (o. J.):
Phonem, http://www.wissen.de/lexikon/phonem, Abruf: 25.12.2014

Kronauer Software (o. J.):
NEU! Dragon NaturallySpeaking 13 Professional – Spracherkennung für Profis, http://kronauersoftware.de/, Abruf: 15.01.2015

Linguatec (2014): Grundlagen der Spracherkennung, http://www.linguatec.de/products/stt/information/basics, Abruf: 25.11.2014

Maher, T./ Arciniega, J. (o. J.):
History of Speech & Voice Recognition and Transcription Software, http://www.dragon-medical-transcription.com/history_speech_recognition.html, Abruf: 15.01.2015

Nuance Communications Inc. (o. J.a):
Dragon NaturallySpeaking Premium, http://www.nuance.de/for-individuals/by-product/dragon-for-pc/premium-version/index.htm, Abruf : 15.01.2015

Nuance Communications Inc. (o. J.b):
Frequently Asked Questions, http://www.nuance.com/for-business/by-product/dragon/product-resources/frequently-asked-questions/index.htm, Abruf : 15.01.2015

Nuance Communications Inc. (o. J.c):
Dragon NaturallySpeaking für das Versicherungswesen, http://www.nuance.de/for-business/by-industry/insurance/Insurance/index.htm, Abruf : 15.01.2015

o. V. (o. J.a): Hidden-Markov-Modell, http://www.fh-wedel.de/~si/seminare/ss01/Ausarbeitung/a.sprache/gdlgsprerk34.htm#3, Abruf: 18.12.2014

o. V. (o. J.b): What is HTK?, http://htk.eng.cam.ac.uk, Abruf: 18.12.2014

o. V. (o. J.c): Vorgegebene Texte aufnehmen und Aufnahmen hochladen, http://www.voxforge.org/de, Abruf: 06.01.2015

o. V. (o. J.d): Willkommen, http://www.voxforge.org/de, Abruf: 06.01.2015

o. V. (o. J.e): What is GPL?, http://www.voxforge.org/home/docs/faq/faq/what-is-gpl, Abruf: 06.01.2015

o. V. (o. J.f): About VoxForge, http://www.voxforge.org/home/about, Abruf: 06.01.2015

o.V. (o. J.g): Nuance Dragon Naturally Speaking 13.0 Premium, http://diktiersoftware.org/dragon-naturally-speak-ing/#Nuance_Dragon_Naturally_Speaking_130_Premium_Wireless, Abruf: 15.01.2015

Petersen W. (2004): Spracherkennung und Hidden Markov Modelle, http://user.phil-fak.uni-duesseldorf.de/~petersen/Einf_CL/Handout_hmm.de.pdf, Abruf: 14.12.2014

Riedel, J. u.a. (2014): Dragon NaturallySpeaking, http://de.wikipedia.org/wiki/Dragon_NaturallySpeaking, Abruf: 15.01.2015

Roodsari, A. V. (2014): Creepy aber hilfreich: So funktioniert Googles persönlicher Assistent, http://www.focus.de/digital/multimedia/googles-now-im-test-creepy-aber-hilfreich-so-funktioniert-googles-persoenlicher-assistent_id_3966860.html, Abruf: 25.11.2014

RWTH ASR (2014): RWTH ASR - The RWTH Aachen University Speech Recognition System, http://www-i6.informatik.rwth-aachen.de/rwth-asr/, Abruf: 20.01.2015

Simon listens (2012a): Funktionsbeschreibung, http://www.simon-listens.org/index.php?id=273, Abruf: 18.12.2014

Simon listens (2012b): Julius, http://simon-listens.org/index.php?id=130, 18.12.2014

Simon listens (2012c): simon, http://www.simon-listens.org, Abruf: 18.12.2014

Spehr, M. (2014): Spracherkennung Dragon 13 – Endlich versteht uns der Computer perfekt, http://www.faz.net/aktuell/technik-motor/computer-internet/gesagt-getan-sprachprogramm-dragon-13-im-praxistest-13128598.html, Abruf: 15.01.2015

Statista (2015): Anzahl der Smartphone-Nutzer in Deutschland in den Jahren 2009 bis 2014 (in Millionen), http://de.statista.com/statistik/daten/studie/198959/umfrage/anzahl-der-smartphonenutzer-in-deutschland-seit-2010/, Abruf: 17.01.2015

WeltN24 GmbH (2014): Einwohnerzahl Deutschlands steigt auf 80,8 Millionen, http://www.welt.de/politik/deutschland/article123649831/Einwohnerzahl-Deutschlands-steigt-auf-80-8-Millionen.html, Abruf: 25.11.2014



Bewertung und Vergleich von Open Source Tools zur Unterstützung leichtgewichtiger Softwareentwicklung

Schriftliche Ausarbeitung
im Rahmen der Lehrveranstaltung „Integrationsseminar“
für das Kompetenzzentrum Open Source (KOS)

Vorgelegt von

Maximilian Rott,
Kerstin Sonnenholzer,
Phuong-Ha Tran

am 23.01.2015

Fakultät Wirtschaft
Studiengang Wirtschaftsinformatik
WWI2012V

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

„Agile Methoden sind gekommen, um zu bleiben. Die hohe Zufriedenheit mit Scrum, Kanban und Extreme Programming zeigt, dass sich das Management von Entwicklungsprozessen und Projekten nachhaltig verändern wird“

Prof. Dr. Ayelt Komus

Professor für Organisation und Wirtschaftsinformatik

Aus diesem Zitat von Professor Doktor Komus ist erkennbar, dass agile Methoden nicht nur ein Trend, sondern eine Verbesserung der klassischen Vorgehensweise mit sich bringen und immer mehr Verwendung in der Softwareentwicklung finden. Laut einer Studie der Firma Anecon aus dem Jahr 2011, werden im Bereich des Testens und der Qualitätssicherung inzwischen bereits in ein Viertel aller Projekte agile Methoden eingesetzt. Hierbei ist der Favorit mit einem Anteil von 57% die agile Methode Scrum.[1]

Agile Methoden bilden einen starken Kontrast zu den klassischen Vorgehensmodellen. Sie laufen nicht in der gleichen sequentiellen Weise ab wie klassische Modelle, sondern in kleineren Iterationen, durch die die Qualität des Produktes früher gesichert wird.

Für die agilen Methoden werden spezielle Werkzeuge benötigt, die die Vorgänge darin optimal darstellen. Bei diesen Werkzeugen für agile Methoden gibt es bereits einen großen Markt, der sowohl mit kommerzieller, als auch mit Open Source Software gefüllt ist. Diese Tools variieren in ihrer Funktionalität und in ihrem Umfang stark, sodass nicht klar erkennbar ist, welches Werkzeug für ein Unternehmen am besten geeignet ist.

Ziel dieser Arbeit ist es, eine Ausschau über den Markt der Open Source Werkzeuge für agile Softwareentwicklung zu schaffen und einen Überblick über die besten Werkzeuge zu geben.

Nach Absprache mit dem Auftraggeber wurde festgelegt, dass in dieser Arbeit nur Open Source und grundsätzlich kostenlosen Werkzeuge betrachtet und bewertet werden sollen. Eine weitere Arbeit, deren Schwerpunkt sich auf die kommerziellen Werkzeuge bezieht, wurde im Rahmen des Projektes von einem anderen Team erstellt.

[1] Haberl, P./Spillner, A./Vosseberg, K. (2012), S. 9-10

2 Theoretische Grundlagen

Um dem Leser einen verständlicheren Einstieg für den Praxisteil zu ermöglichen, wird im Folgenden die Begriffe Marktanalyse und Open Source erklärt. Die agilen Methoden Scrum und Kanban, welche das Tool später unterstützen soll, werden im Kapitel 2.3 genauer betrachtet und ihre Unterschiede herausgestellt.

2.1 Marktanalyse

Um den Markt transparent zu gestalten, damit beispielsweise Vergleiche bezogen auf den Erfolg von Unternehmen, unterschiedlicher Produkte oder ähnliches geschaffen werden können, gibt es die sogenannte Marktanalyse. Die zu beobachtenden Bereiche unterscheiden sich bei der Marktanalyse je nach dem Ziel und was verglichen wird. Beispielsweise hat ein Unternehmen das Ziel herauszufinden wie erfolgreich es im Vergleich zu andern Unternehmen ist. Eine Marktanalyse findet in einem festgelegten Zeitraum statt und wendet häufig Statistiken oder Meinungsforschungen an.[2]

In dieser Projektarbeit liegt der Schwerpunkt auf der Auswahl eines geeigneten Tools. Hierbei redet man von einer sogenannten Angebotsanalyse. Bevor eine Angebotsanalyse durchgeführt werden kann, müssen Angebote eingeholt werden.

Hierzu werden grundlegende Anforderungen an die Beschaffenheit und Leistung des Produkts gestellt. Bei diesem Vorgehen wird auch ein hoher Wert auf die Leistung des Lieferanten und dessen Fähigkeit gelegt.[3]

Die heutigen Möglichkeiten, um Angebote für Produkte einzuholen, sind durch das Internet exponentiell angestiegen. Auch in dieser Projektarbeit wird die Angebotsbeschaffung über eine Homepage durchgeführt, auf der mehrere Tools aufgeführt werden.

Hierbei kann es zu Risiken durch Globalen Anbieter kommen. Zu diesen Risiken gehören Politische unterschiede, Währungsdifferenzen, Unterschiedliche Auffassung der Produktqualität und Risiken bzgl. des Vertragsrechts.[4]

Wurden einige Angebote eingeholt, folgen die Angebotsanalyse und ein Angebotsvergleich welche dazu dienen die eingeholten Produkte nochmals auf die benötigten Anforderungen zu überprüfen und miteinander zu vergleichen.[5]

[2] Wübbenhorst, K. (o. J.)
[3] Schuh G./Schmidt C. (2014), S. 240 - 241
[4] Deutsches Institut für Interne Revision e.V. (2010), S. 46 ff.
[5] Schuh G./Schmidt C. (2014), S. 240 - 241

Die Angebotsanalyse betrachtet jedes Produkt einzeln und in die Tiefe. Der Angebotsvergleich stellt anschließend diese nebeneinander und vergleicht die Produkte untereinander. Um Dritten zu ermöglichen diesen Prozess später nachzuvollziehen ist es besonders wichtig den Angebotsvergleich detailliert zu dokumentieren.

Zu den geläufigsten Vergleichsfaktoren zählen der Preis, das Qualitätsniveau, die Lieferzeit, die Zuverlässigkeit, die Kapazität, der Service und der Standort. Je nach Produkt oder Dienstleistung variieren diese Faktoren und Ihre Gewichtung.[6] Bei einem Tool wird zum Beispiel ein hoher Wert auf den Support und somit den Service sowie auf die Zuverlässigkeit gelegt.

2.2 Open Source

Der Begriff „Open Source" bezeichnet eine Art von Software, deren Quellcode öffentlich zugänglich ist und frei verändert werden darf.[7]

Der Begriff entstand Ende der 90er Jahre aus der „Open Source Definition" (dt. „Offene Quellen Definition"), die beschreibt welche Bedingungen eine Software haben muss, die sich als „Open Source" bezeichnet.[8] Eine „Open Source"- Anwendung muss einen frei zugänglichen Code besitzen, muss kostenlos sein, und muss verändert und angepasst werden dürfen.

„Open Source" muss abgegrenzt werden von den Begriffen „Freeware", „Public Domain Software" und „Shareware", da diese keine Aussage über die Zugänglichkeit des Quellcodes, sondern über die Erlaubnis der Verteilung und Vermehrung beschreiben. Aus Tabelle 1, welche im Folgenden dargestellt ist, sind die Definitionen der unterschiedlichen Softwarearten zu erkennen.

[6] Büsch, M. (2013), S. 179
[7] Haun, M. (2014), S.417
[8] Fox, D. (2006)

Public Domain Software	Kostenlose Software, die frei vermehrt und verändert werden darf und keine Besitzansprüche besitzt.
Freeware	Kostenlose Software, die nur bedingt vermehrt und meistens nicht verändert werden darf.
Shareware	Kommerzielle Software, die für einen begrenzten Zeitraum oder Umfang verwendet und vermehrt werden darf. Eine gekaufte Version darf nicht vermehrt oder verändert werden.

Tabelle 1 Definition Softwarearten[9]

Die Begriffe „Public Domain Software“ und Open Source werden im geläufigen Sprachgebrauch häufig miteinander vermischt oder verwechselt. Der Unterschied zwischen „Public Domain Software“ und „Open Source“ ist, dass bei „Open Source“ beim Ersteller immer noch die Besitzansprüche bestehen und dieser die Verbreitung und Veränderung erlaubt.

Es gibt nach der „Open Source Definition“ auch bestimmte Voraussetzungen, durch die eine Lizenz als „Open Source Lizenz“ bezeichnet werden darf. Deshalb gibt es auch nicht nur eine Lizenz, unter der alle „Open Source“- Anwendungen laufen.

Die wichtigsten Lizenzen sind die „General Public License“ (GPL), die „Apache License“, die „BSD-Lizenz“ und die „Common Public License“ (CPL).[10]

2.3 Leichtgewichtige Methoden

Vorgehensmodelle beschreiben unterschiedliche Regelwerke zur Durchführung eines Projektes[11]. Sie stellen Elemente und Methoden des Projektmanagements zu Prozessen und Phasen innerhalb eines Projektablaufes zur Verfügung. Agile Vorgehensmodelle gelten als leichtgewichtig. Dies bedeutet, dass der funktionierende Code und nicht die Dokumentation im Vordergrund steht. Agile Vorgehensmodelle sind nicht unstrukturierter als strengere Phasenmodelle, sie legen nur eine andere Priorisierung gegenüber anderen Vorgehensmodellen fest.[12]

[9] Vgl. Haun, M. (2014), S.417
[10] Vgl. ifross (o.J)
[11] Vgl. IT Wissen (2014)
[12] Vgl. Hanser, E. (2010), S. 3 ff.

Agile Vorgehensmodelle enthalten einzelne Elemente zu Techniken und Methoden, die auf ein Projekt baukastenartig angewendet werden. Die Elemente bündeln Entwicklungsmaßnahmen, welche mit leichtgewichtigen Vorgehensweisen eine höhere Flexibilität ermöglichen.[13]

Die Vorlage für agile Vorgehensmodelle ist das "Agile Manifest".[14] Agil bedeutet „von großer Beweglichkeit zeugend; regsam und wendig".[15] Das „Agile Manifest" hat diese Bedeutung verankert. Der Leitsatz des „Agilen Manifests" priorisiert die individuellen Beteiligten und ihre Interaktion, wodurch die gewünschte, lauffähige Software und eine gute Zusammenarbeit mit dem Kunden erreicht werden.

Der Leitsatz des „Agile Manifests" lautet wie folgt:

„Wir erschließen bessere Wege, Software zu entwickeln, indem wir es selbst tun und anderen dabei helfen. Durch diese Tätigkeit haben wir diese Werte zu schätzen gelernt:

Individuen und Interaktionen mehr als Prozesse und Werkzeuge
Funktionierende Software mehr als umfassende Dokumentation
Zusammenarbeit mit dem Kunden mehr als Vertragsverhandlung
Reagieren auf Veränderung mehr als das Befolgen eines Plans

Das heißt, obwohl wir die Werte auf der rechten Seite wichtig finden,
schätzen wir die Werte auf der linken Seite höher ein."[16]

Der Vorteil des agilen Vorgehens liegt für Kunden und Entwickler in der Möglichkeit, Veränderungswünsche im Laufe des Projekts zu äußern und diese nach finanziellen und technischen Gegebenheiten umzusetzen. Aus diesem Grund sind diese Vorgehensmodelle für kleine Teams sinnvoll, bei denen die Anforderungen anfangs nur unvollständig bekannt sind.[17] Ein Beispiel für agile Vorgehensmodelle ist „Scrum".[18]

2.3.1 Scrum

Scrum entstand während der 1990er Jahre, als in dieser Zeit leichtgewichtige Methoden immer populärer wurden. Es hat sich herausgestellt, dass kleine Teams effektiver arbeiten und ein qualitativ höherwertiges Ergebnis präsentieren können – sofern alle Projektmitglieder auch nahezu identische Fähigkeiten besitzen. Diese Methode hat sich laut Gloger deshalb

[13] Vgl. Kuhrmann, M. (2013)
[14] Vgl. Hanser, E. (2010), S. 7
[15] Duden (o. J.a)
[16] Beck, K./Beedle, M./van Bennekum, A. (2001)
[17] Vgl. Hanser, E. (2010), S. 3 ff.
[18] Vgl. Kleuker, S. (2013), S. 24

als Standard der agilen Softwareentwicklung durchgesetzt, weil das Management und Business angesprochen wird. Scrum kann mit jeder gängigen Softwareentwicklungsmethode kombiniert werden.[19] Scrum ist ein Rahmen für die Entwicklung und Erhaltung komplexer Produkte. Diese agile Projektmanagementmethode wurde von Ken Schwaber und Jeff Sutherland entwickelt.[20]

Scrum ist kein Prozess und keine Technik, um Produkte zu bilden, es ist vielmehr eine Grundstruktur, welches unterschiedliche Prozesse und Techniken nutzt. Scrum verdeutlicht die relative Effizienz des Produktmanagements und Entwicklungspraktiken, welche man verbessern kann.

Die Grundstruktur besteht aus Scrum Teams und deren assoziierten Rollen, Ereignisse, Produkte und Gesetze. Jede Komponente innerhalb dieses Grundgerüsts dient einem besonderen Zweck und ist essentiell für den Erfolg und Gebrauch.

Die Gesetze von Scrum verbinden die Ereignisse, Rollen und Produkte, steuern die Beziehungen und Wechselbeziehungen zwischen einander.

Das Scrum Team besteht aus einem Product Owner, dem Development Team und einem Scrum Master. Hier wird zwischen Product Owner und Scrum Master getrennt, da unterschiedliche Interessen bestehen, die im Widerspruch stehen. Ein Beispiel dafür wäre, dass der Projekt Owner möglichst so viel wie möglich, so schnell wie möglich erledigt haben will. Der Scrum Master dagegen achtet darauf, dass keine Überlast für sein Team entsteht. Im klassischen Projektmanagement muss dieser Konflikt vom Projektleiter in einer Person gelöst werden. Diese Scrum Teams organisieren sich selbst und sind übergreifend-funktional. Selbstorganisierte Teams entscheiden selbst, wie sie ihre Ziele am besten erreichen können, besser als außerhalb des Teams von anderen gelenkt zu werden.

Scrum Teams sollten alle benötigten Kompetenzen haben, um ihre Ziele zu erreichen, ohne sich auf andere außerhalb des Teams verlassen zu müssen. Diese Teamstruktur in Scrum ist so gestaltet, um die Flexibilität, Kreativität und Produktivität zu optimieren.

Scrum Teams liefern Produkte in schrittweise regelmäßigen Abständen, um die Möglichkeit für Feedback zu maximieren. Schrittweise Lieferungen von abgeschlossenen Produkten gewährleisten einen potenziellen Nutzen, um die Qualität der einzelnen Versionen zu gewährleisten. Die nachfolgende Grafik (Abb. 1) soll einen Einblick in den Ablauf und Funktionsweise von Scrum verdeutlichen.

[19] Vgl. Gloger, B. (2010), S. 195-196
[20] Vgl. Scrum Guides (2014)

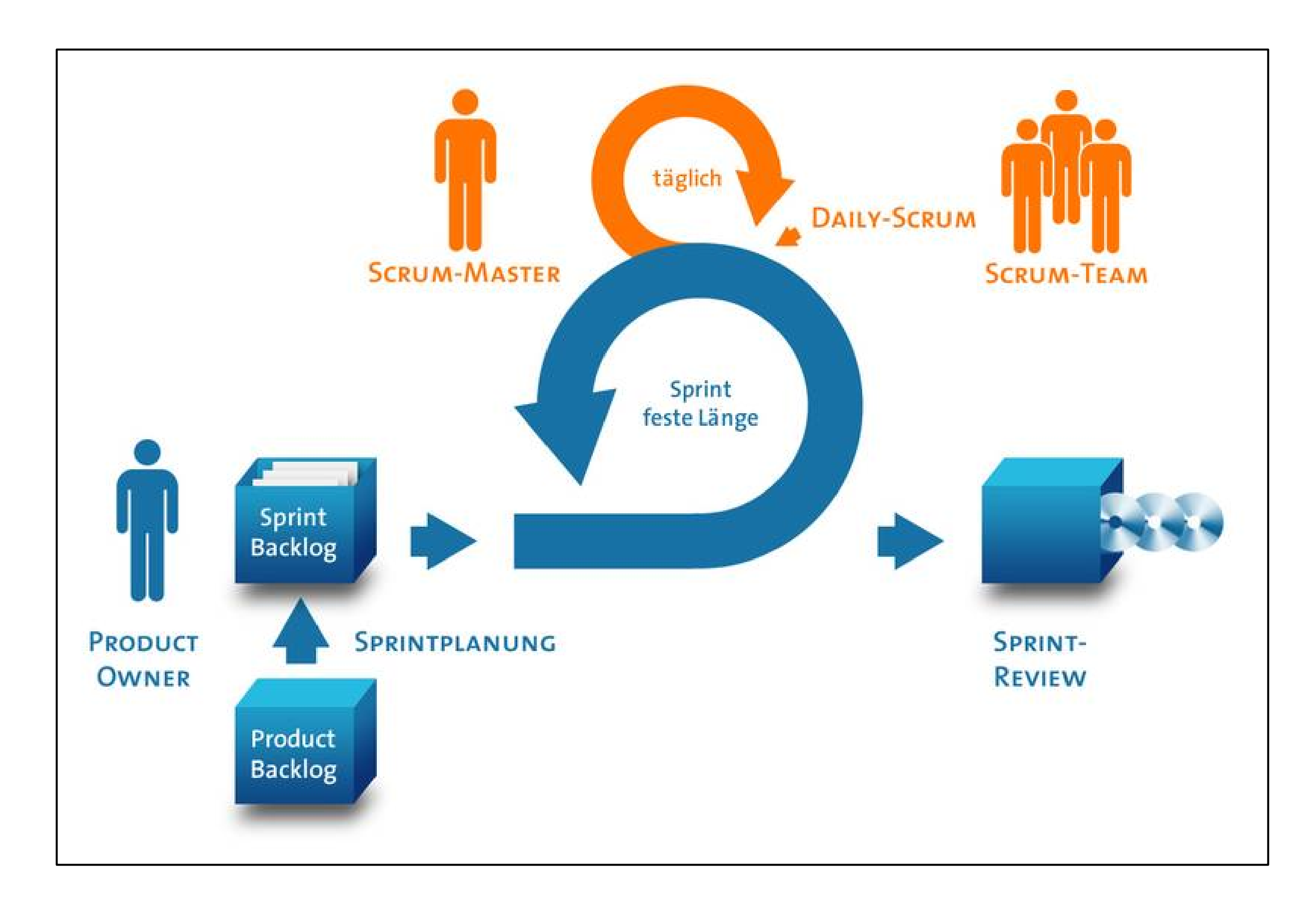


Abb. 1: Prozessablauf Scrum[21]

In Scrum ist der Product Owner für ökonomischen Erfolg des Produktes verantwortlich. Er beginnt die Produktentwicklung mit einer klaren Produktvision. Passend zur Produktvision, erstellt und priorisiert er die Anforderungen im Product Backlog. Er bindet alle relevanten Beteiligten früh in die Definition der Produktvision und des Product Backlogs ein. Der Product Owner bleibt aber stets der Verantwortliche.

Zu Beginn jedes Entwicklungszyklusses, welcher in Scrum Sprint genannt wird, findet das Sprint Planning statt. Beim Sprint Planning vereinbaren der Product Owner und das Entwicklungsteam, welche Anforderungen aus dem Product Backlog im aktuellen Sprint erledigt werden sollen. Diese Anforderungen (auch User Stories genannt) werden in das Sprint Backlog übertragen. Es ist essentiell, dass sich beide Parteien über den Inhalt des Sprint Backlog einig sind. Das Entwicklungsteam kann hierbei selbst einschätzen, wie viel es bis zum nächsten Sprint erledigen kann.

Ein Sprint hat eine feste, immer gleiche Länge. Die Entwicklung erfolgt innerhalb von direkt aufeinanderfolgenden Sprints. Die gleichbleibende Länge erleichtert einen Vergleich der unterschiedlichen Sprints. Insbesondere aus der Geschwindigkeit der bisherigen Sprints kann sehr genau die Geschwindigkeit der nächsten Sprints prognostiziert werden. Viele Teams

[21] Enthalten in: 3m5 (2011)

arbeiten in Intervallen von zwei oder drei Wochen als feste Sprintlänge, aber auch eine oder vier Wochen sind nicht unüblich.

Das selbstorganisierte Team arbeitet während der Sprints störungsfrei an der Abarbeitung des Sprint Backlogs. Dafür zu sorgen ist eine der Aufgaben vom Scrum Master, auch wenn der Projekt Owner was anderes will. Innerhalb dieser Zeit werden auch keine neuen oder geänderten Anforderungen bearbeitet. Dies ist erst nach dem Sprint im Sprint Planning des nächsten Sprints möglich.

Das Daily Scrum sorgt für den täglichen Infoaustausch und die Einsatzplanung für den Tag. Es findet jeden Werktag zur selben Uhrzeit statt und dauert maximal 15 Minuten. Daher wird es meist im Stehen durchgeführt. Alle Teilnehmer beantworten folgende Fragen:

1. Was habe ich seit dem letztem Daily Scrum gemacht?
2. Was behindert mich bei meiner Arbeit?
3. Was werde ich bis zum nächsten Daily Scrum erledigt haben?

Das Daily Scrum beinhaltet keine Diskussionen. Diese können meistens in kleineren Gruppen im Anschluss besprochen werden.

Der Scrum Master sorgt dafür, dass Scrum im Team funktioniert. Er kümmert sich darum, dass der Scrumprozess eingehalten wird, hilft bei der Beseitigung von Hindernissen, schützt das Team gegen Störungen von außen und hilft dem Team bei der Selbstorganisation. Er löst dabei nicht direkt die Probleme des Teams, sondern hilft dabei, die Probleme selbst zu lösen. Der Scrum Master kann zu einem gewissen Grad mit einer Führungskraft gleichgestellt werden. Er ist aber gegenüber dem Team nicht weisungsbefugt und folgt dem Grundsatz „führen durch dienen".

Am Ende jedes Sprints findet das Sprint Review statt. Hier präsentiert das Team dem Product Owner seine Fortschritte. Dafür ist es wichtig, dass die Anforderungen im Sprint Planning so heruntergebrochen werden, dass es nach jedem Sprint etwas vorzuführen bzw. zu sehen gibt. Häufig fallen den Projektbeteiligten im Sprint Review neue Anforderungen ein, die der Product Owner direkt in das Product Backlog einstellt.

Die Fortschrittsanalyse erfolgt innerhalb der Sprints anhand eines Burndown Charts (Abb. 2). Das Burndown Chart gibt Auskunft über die noch zu leistende Arbeit. Auf der x-Achse werden Tage angegeben und auf der y-Achse die Arbeit in Stunden. Jedes Teammitglied schätzt täglich den Restaufwand seiner Tasks, somit ergibt sich eine von Tag zu Tag verbesserte Genauigkeit. Hieraus können in gewissem Maße Trends und Prognosen abgeleitet werden.

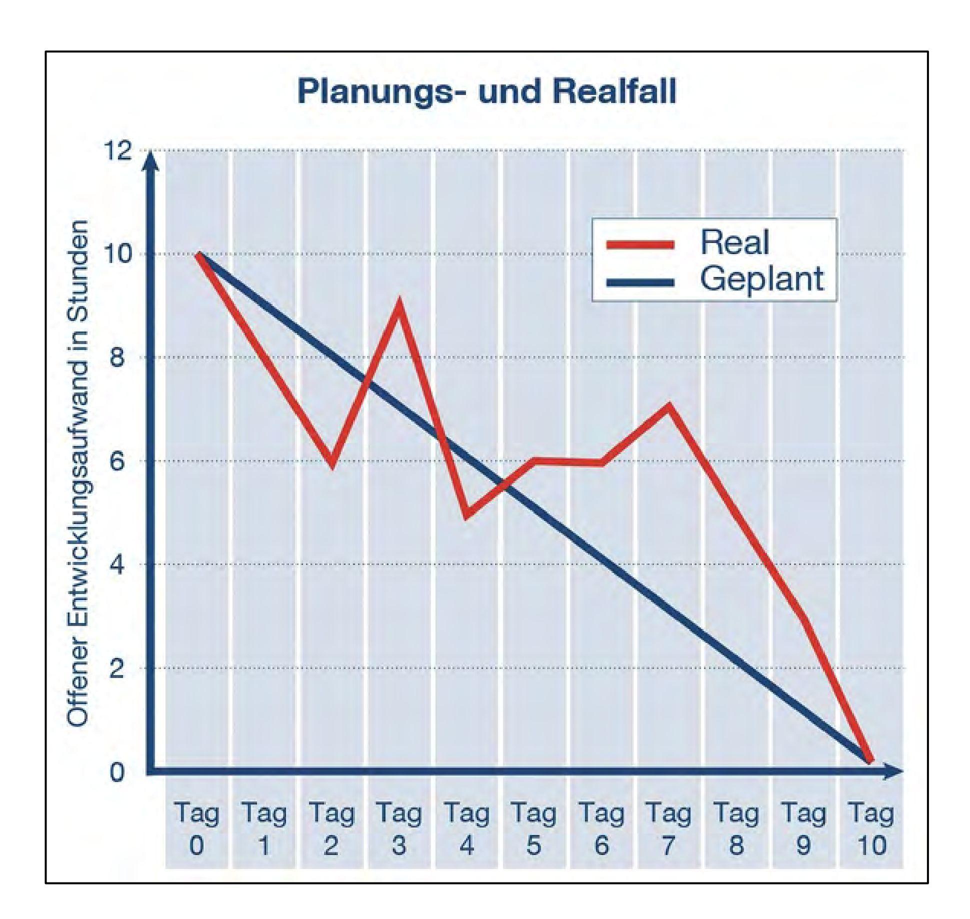


Abb. 2: Burndown Chart[22]

Während es im Sprint Review um das Betrachten des Ergebnisses geht, betrachten Scrum Teams in der Retrospective ihren Prozess. Sie fragen sich einerseits, was gut gelaufen ist und wie diese positiven Erfahrungen auch auf zukünftige Projekte übertragen werden können. Andererseits betrachten sie auch die negativen Seiten und wie diese verbessert werden können. Dies sichert den kontinuierlichen Verbesserungsprozess. Anschließend folgt der nächste Sprint.[23]

Außer Scrum gibt es noch einige andere agile Methoden, um den Ablauf eines Projektes zu steuern. Im folgenden Kapitel wird Kanban als weitere Möglichkeit vorgestellt.

2.3.2 Kanban

So wie Scrum auf dem Agilen Manifest und dieses im Kern auf „Lean Management“ beruht, beruht Kanban direkt auf Lean Management. Die Übersetzung des Begriffes Lean ist schlank oder mager und erzeugt Flexibilität. Ein schlankes Unternehmen konzentriert sich auf das Wesentliche und verbraucht für deren Erfüllung von allem weniger. Weniger Zeit, weniger

[22] Enthalten in: Scrum Kompakt (2014)
[23] Vgl. it-agile (2014)

Ressourcen. Im Lean Management geht es deshalb im Kern darum, mit weniger Personal, weniger Arbeitsfläche, weniger in Infrastruktur/Equipment gebundenen Mitteln und kürzerer Entwicklungszeit trotzdem hervorragende Produkte bereitzustellen.[24]

Das Prozessmanagement System Kanban kommt ursprünglich aus Japan und dem Bereich der Fertigungssteuerung. Die Methode wurde ca. 1940 in der Automobilindustrie von Toyota entwickelt. Es ist das Ziel, die Durchlaufzeiten bei der Produktion von Automobilen zu verringern.[25]

In der Produktionsplanung und -steuerung oder allgemein Logistiksteuerung wird mit dem Produktions-Kanban ein Steuerungsverfahren beschrieben, mit dem verbrauchtes Material nach dem Verbrauchssteuerungsprinzip (Pull-Prinzip) wieder neu befüllt bzw. versorgt wird. Die Materialbereitstellung bei Kanban orientiert sich damit ausschließlich am Verbrauch im Produktionsablauf. Kanban ermöglicht eine nachhaltige Reduzierung der Materialbestände und Erhöhung der Lieferfähigkeit.

Kanban-Systeme steuern im Idealfall die gesamte Wertschöpfungskette vom Lieferanten bis zum Endkunden. Auf diesem Weg können Versorgungsabrisse weitgehend vermieden werden und Überbestände sicher ausgeschlossen werden. Die Kanban-Steuerung benötigt allerdings eine kontinuierliche Überwachung auf Eignung jedes Materials.[26]

Die Steuerung des Materialflusses erfolgt durch Behälter, indem die Rohstoffe enthalten sind und durch Kanban-Karten. Die Kanban-Karten dienen als ein Identifikationsmittel an den Behältern und haben den Vorteil, dass zum Anstoß einer Nachlieferung nur die physische Karte oder die elektronische Information der Karte (eKanban) transportiert werden muss. Sehr beliebt bzw. häufig zu finden ist eine Kanban-Steuerung mit zwei Behältern. Während sich ein Behälter in der Produktion befindet ist der zweite Behälter unterwegs beim Nachfüllen. Wenn das letzte Teil aus dem Behälter entnommen wird, entfernt man die Kanban-Karte vom Behälter und fügt sie in den Briefkasten wo alle neuen Kanban-Karten gesammelt werden. Zwischenzeitlich kommt der Pufferbestand zum Einsatz. Die Mailbox wird geleert, alle Karten werden erfasst, gebucht und sortiert. Danach wird der Auftrag an die Lieferanten oder die Produktion weitergeleitet. Diese füllen das Leergut wieder auf. Die wieder vollen Behälter werden dem Pufferbestand hinzugefügt. Es werden drei Varianten unterschieden, je nachdem, ob nur aus einem Lager bereitgestellt und kommissioniert (Lager-Kanban), bestellt (Lieferanten-Kanban) oder nachproduziert (Produktions-Kanban) wird.

[24] Vgl. LEAN Managementmethoden (2015)
[25] Vgl. WinfWiki (2014)
[26] Vgl. Dickmann, P./Dickmann, E. (2014)

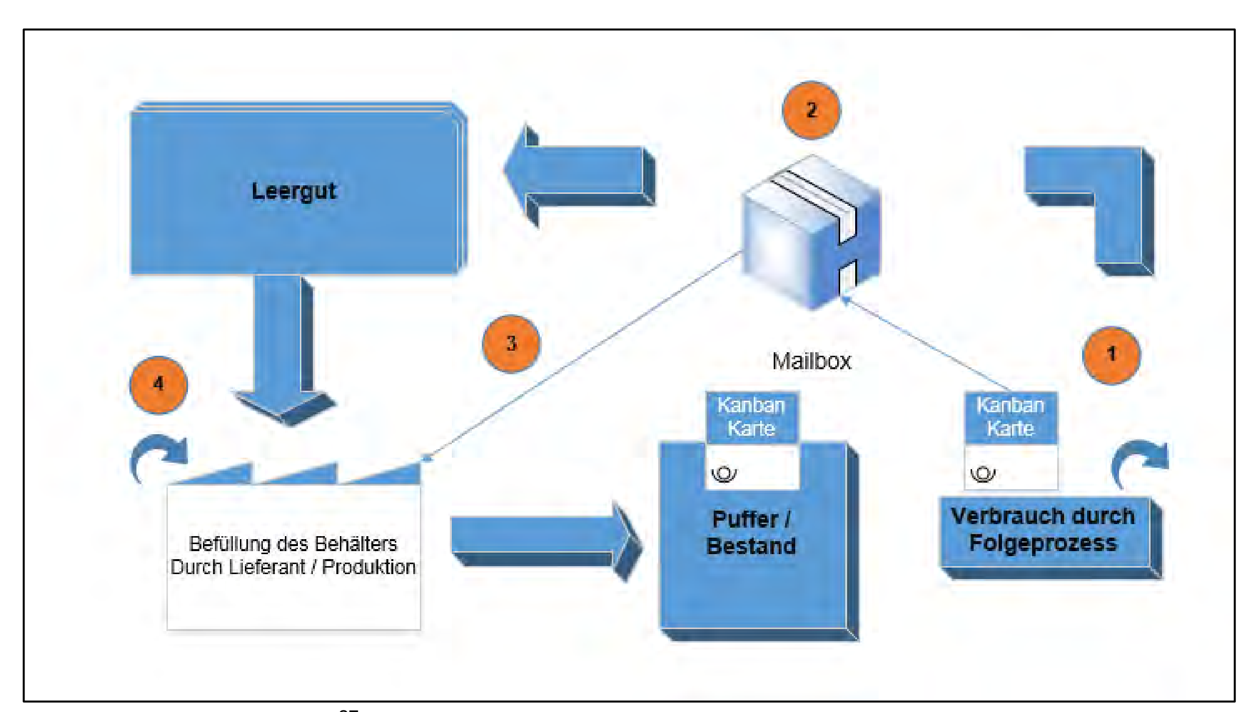


Abb. 3: Kanban Prozess[27]

Kanban lässt sich generell in den meisten Fertigungsbereichen einsetzen, unabhängig davon ob die Teile direkt bereitgestellt, aus dem Lager entnommen oder gefertigt werden.

Sehr gut geeignet ist Kanban für die einstufige Produktionstiefe (z.B. Endproduktfertigung) wie in der Abbildung 3 gezeigt. Die Reduzierung von vorhandenen Beständen ist damit jedoch gering. Bei wenigen Einzelteilen ist klassisches Karten-Kanban möglich. Es bietet sogar die Möglichkeit bei der Produktion wie bei Toyota ohne aufwändiges MRP-System auszukommen. Ebenso eignet sich Kanban sehr gut für die Serienproduktion und einstufige Produktionsprozesse (z.B. Metallverarbeitung). Hier lassen sich in der Regel auch deutliche Bestandsreduzierungen und Erhöhung der Lieferfähigkeit erreichen. Allerdings sollte bei der Serienproduktion darauf geachtet werden, dass möglichst alle Teile über Kanban versorgt werden.

Klassisches Kanban ist ungeeignet für Einzelfertigung und Sonderprodukte, und den Einkauf von den dazu notwendigen Rohmaterialien und Teilen mit sporadischen, punktuellen Bedarf. Soll hier ebenfalls Kanban zum Einsatz kommen, sollte hybrides MRP-Kanban verwendet werden.[28]

Kanban findet auch in IT Bereichen Einsatz. Im Folgenden ist mit Kanban das SW-Kanban (Software-Kanban) gemeint. Kanban basiert wie agile Methoden auf dem Lean Management.

[27] Mit Änderungen entnommen aus: .Versicherung (2011), S. 6
[28] Vgl. Dickmann, P./Dickmann, E. (2014)

Es zielt auf die Verbesserung des Prozesses ab, wobei die Teamarbeit und die Mitarbeiter selbst nicht im Vordergrund stehen. Die wichtigsten Prinzipien lauten wie folgt:

1. „Ausrichtung an realen Bedarfen“
2. „Dezentrale Steuerung“
3. „Vermeidung von Verschwendung“
4. „Methodenmix“
5. „Kontinuierliche Verbesserung“[29]

An dieser Stelle ist auch wichtig zu beleuchten, welche Aktionen unterlassen werden sollten.

- Einerseits sollte der bestehende Prozess nicht grundlegend verändert werden, weil es zu viel Zeit kostet, neue Prozesse einzuführen und diese nicht immer zu Akzeptanz führen.
- Auch das Vorgehen der Mitarbeiter bei der Durchführung ihrer Arbeit sollte nicht abgeändert werden, weil der Mitarbeiter mit einer neuen Vorgehensweise nicht vertraut ist und dies wiederum einiges verzögern kann.
- Andererseits soll die Performance der Mitarbeiter nicht angezweifelt, herausgestellt oder gar zum Kernpunkt gemacht werden. Der Mitarbeiter wird es als eine Kritik gegen die eigene Persönlichkeit sehen und so eher kontraproduktiv arbeiten.

Bei Kanban wird der Workflow visualisiert. Die einzelnen Bearbeitungsstationen einer Anforderung werden innerhalb einer Tabelle dargestellt. Diese Tabelle hat unterschiedliche Spalten für die vorhanden Aufgaben (Story Backlog), die noch zu erledigenden Anforderungen (Requirements), die gerade im Entwicklungsstatus befindlichen Anforderungen (Development), die gerade im Test befindlichen Anforderungen (Test), die im Implementierungsprozess befindlichen Tasks (In Production). Es gibt unterschiedliche Gestaltungsmöglichkeiten des Kanban Boards. Dies ist nur ein Beispiel dafür.

[29] Vgl. Dickmann, P. (2007), S. 185

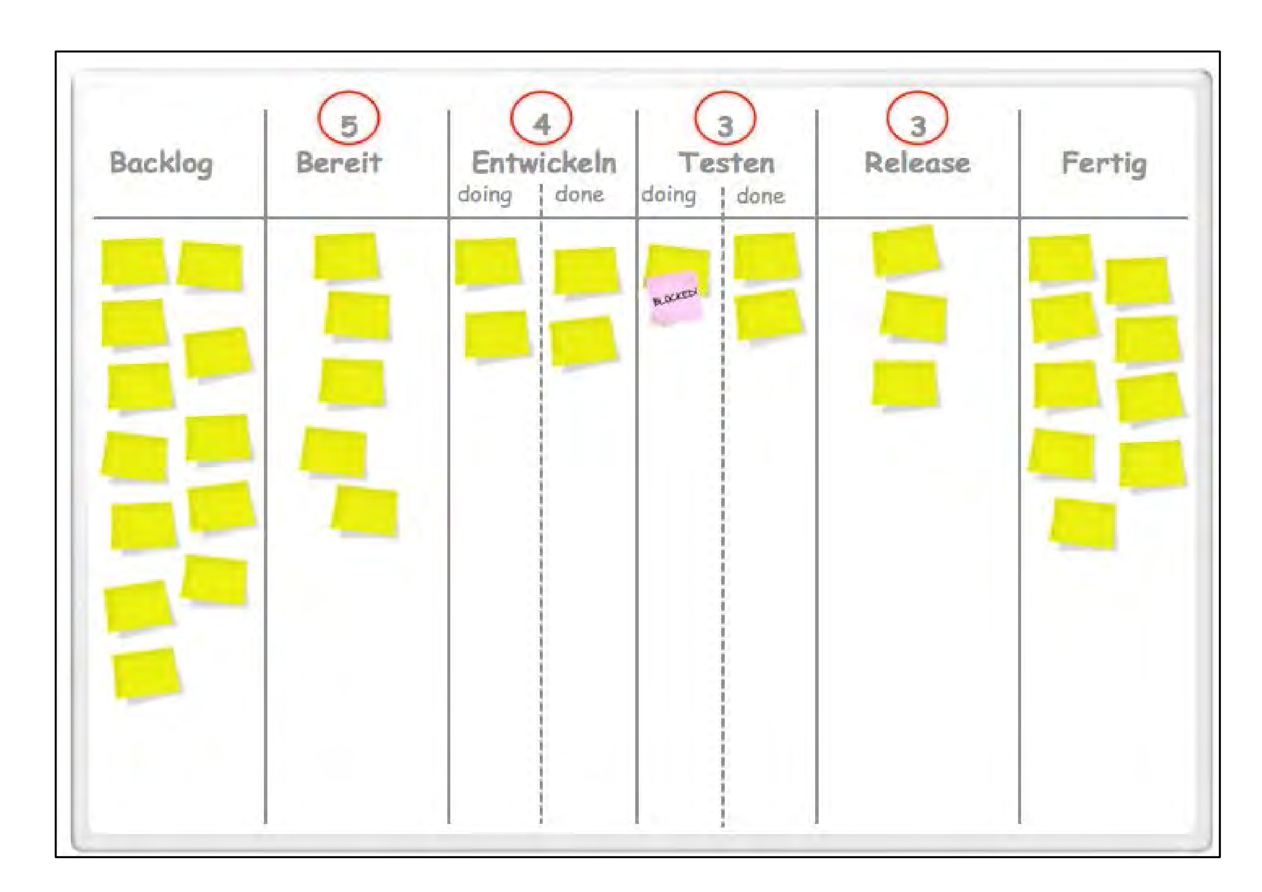


Abb. 4: Kanban Board[30]

Die Menge an paralleler Arbeit (Work in Progress: WiP) wird im nächsten Schritt beschränkt. Wie in Abbildung 4 dargestellt, wird für jeden Prozessschritt die maximale Anzahl an Anforderungen festgelegt, die parallel in diesen Prozessschritt bearbeitet werden darf.

Durchlaufzeiten (auch lead time) werden gemessen und optimiert. Die durchschnittliche Durchlaufzeit einer Anforderung wird in diesem Zuge ermittelt. Diese Durchlaufzeit wird mittels WiP-Werte so optimiert, dass diese möglichst kurz und voraussagbar ist.

Der erste Schritt, um einen Kanban Board zu erstellen, ist eine „Value Stream"-Analyse durchzuführen und festzustellen, welcher Teilausschnitt des Prozesses mit Kanban optimiert werden soll. Danach wird für jede Spalte ein Prozessschritt erstellt. Das WiP-Limit je Prozessschritt wird festgelegt, da die Minimierung des WiP-Limits zu einer Erhöhung des Durchflusses führt. Im nächsten Schritt werden die Abnahmekriterien für den Übergang zum nächsten Prozessschritt festgelegt. Die Möglichkeit wird geschaffen, kleine Aktivitäten ihrer übergeordneten Aufgabe zuordnen zu können. Die Aufgaben werden in kleine, möglichst gleichgroße Aktivitäten geordnet.

[30] Enthalten in: it-agile (o. J.)

Das Kanban Team trifft sich täglich und diskutiert, was den Prozessflow behindert. Zusätzlich sollte es regelmäßige Meetings geben, die das Ziel der Prozessoptimierung im Bilde hat.

2.3.3 Unterschiede zwischen Scrum und Kanban

Betrachtet man die verschiedenen Stationen, die zum größten Teil in Scrum benutzt werden, dann erkennt man, das Scrum eine organsiertere Softwareentwicklungsmethode ist.[31] Durch die umfangreichen Stationen lassen sich in Scrum genaue Abläufe für Sprints planen, Stories genau und detailliert beschreiben und priorisieren und Releases durch ein Burndown-Chart auf den Tag genau voraussagen. Kanban hingegen nutzt als einziges Organisations- und Dokumentationsmittel das Kanban-Board. Der große Nachteil dabei ist, dass bei großen Projekten schnell die Übersicht auf dem Kanban-Board verloren gehen kann. Wie Scrum nutzt Kanban ebenfalls ein Burndown-Chart um Release Planungen machen zu können.

Es gibt unterschiedliche Ansätze zur Begrenzung der Arbeitsmenge. Bei Scrum wird die Menge der Tätigkeiten durch Sprints begrenzt. Bei neuen Anforderungen muss bis zum nächsten Sprint gewartet werden. Bei Kanban wird die Menge durch die Anzahl der in Bearbeitung befindlichen Tätigkeiten (WiP) begrenzt. Hier weiß man nicht genau, wann der nächste Platz frei ist.

Ein anderer Unterschied äußert sich im Design des Task-Boards. Scrum an sich adressiert die Produktumsetzung, während Kanban ggf. die gesamte Prozesskette adressiert. In Scrum werden die einzelnen Anforderungen innerhalb der Sprints verarbeitet. Diese Anforderungen orientieren sich hauptsächlich an der Projekt- und Produktumsetzung. Im Kanban Board werden einzelne Prozessteile der Prozesskette beleuchtet.

Eine andere Abgrenzung findet sich in der Prozessgestaltung und der Teamaufstellung. In Scrum ist jedes Team multifunktional besetzt und besitzt ein Backlog, dessen vollumfängliche Bearbeitung zu den Aufgaben des Teams gehört. Zu Kanban gibt es zwei Möglichkeiten. Die eine Variante ist, dass einfache Prozesse nur von einem multifunktionalen Team betreut werden. Die andere Variante bezieht sich auf andere Besetzungen des Personals. Beispielsweise können einigen Bearbeitungsstationen ein Spezialist oder ein Spezialisten Team zugeordnet werden, während andere Bearbeitungsstationen vom multifunktionalen Team abgedeckt werden.

Bei der Aufgabengröße gibt es unterschiedliche Regeln. Scrum Teams fokussieren sich darauf, am Ende eines Sprints Arbeit produktionsreif umgesetzt zu haben. Kanban Teams fokussieren auf die Minimierung der Durchlaufzeit, was dazu führt, dass die Arbeit in kleine, überschaubare Teile herunter gebrochen wird. Kanban hat keine ausdrückliche Regel, dass

[31] Vgl. Roock, S. (2014)

die Arbeit in eine festgelegte Timebox passen muss. Kleinere Aufgaben erhöhen aber den Durchsatz der bearbeiteten Aufgaben.

Weitere Differenzen äußern sich beim Product Backlog. Bei Scrum muss dieses Backlog unbedingt existieren. Änderungen hierzu finden erst im nächsten Sprint statt. Der Product Backlog ist bei Scrum nach der Priorität sortiert. Bei Kanban ist dieser nur optional. Veränderungen zum Produkt treten in Kraft, sobald ein Platz freigeworden ist. Bei der Priorisierung im Kanban Board können alle möglichen Ansätze genommen werden. Die einfachste Variante wäre die Wahl eines zufälligen Tasks. Eine andere Möglichkeit ist immer das älteste Item oder immer die obersten zu entnehmen. Die Wahl der Priorisierung wird dem Team überlassen.[32]

[32] Vgl. mindmeister (2013)

3 Herausforderung

In Unternehmen gewinnen agile Methoden wie Scrum und Kanban immer mehr an Bedeutung. Für diesen Einsatz sind Tools, die diese Methoden unterstützen eine gute und notwendige Hilfestellung.

Auf dem Markt finden sich unzählige Tools zur Unterstützung von agilen Methoden im Softwareentwicklungsprozess. Vor allem bei Open Source Tools sind starke Unterschiede in der Qualität und im Leistungsumfang feststellbar. Im weiteren Verlauf dieser Arbeit werden diese nach unterschiedlichen Kriterien und deren Gewichtung bewertet und verglichen. Der Fokus wird auf Open Source Tools gelegt, ebenfalls werden auch Freeware Tools betrachtet. Dies dient dazu sicher zu stellen das geeignetste Tool auszuwerten.

Vom Auftraggeber werden bestimmte Anforderungen an das später einzusetzende Tool vorausgesetzt. Hierzu gehören insbesondere Marktverbreitung, Reifegrad, Support, sowie Skalierung und Kosten. Das Ziel ist es ein geeignetes Tool für den Einsatz von Scrum und Kanban herauszuarbeiten.

4 Kriterienkatalog als Lösungsansatz

Eine kompetente Auswahl bei dem großen Angebot von Tools zur Unterstützung von Agilen Methoden zu treffen ist eine große Herausforderung. Es ist besonders wichtig die Auswahl der Tools nach einem einheitlichen Vorgehen zu treffen, welches eine ausführliche Dokumentation beinhaltet.

Um zwischen optimalen Tools und Tools die, die Anforderungen nicht erfüllen zu unterscheiden wird ein Kriterienkatalog erstellt. Dieser umfasst die Anforderungen des Auftraggebers, die von den Autoren erweitert und genauer beschrieben werden. Die einzelnen Anforderungen haben eine unterschiedliche Gewichtung, die vor einer Bewertung der Tools getroffen werden muss. Einige sind notwendig, wohingegen andere mit geringerer Gewichtung zu betrachten sind. Eine Skala von 1-10, wobei 10 für am wichtigsten steht, zeigt an wie relevant die Anforderungen beim Auszuwählenden Tool sind.

Im ersten Schritt werden die Tools nach einem K.O.-Auswahlverfahren aussortiert, um anschließend diese ausgewählten Tools in den Kriterienkatalog einzutragen und miteinander zu vergleichen. Der Kriterienkatalog beinhaltet die festgelegt und gewichteten Kriterien.

Ein Kriterienkatalog hat somit die Möglichkeit eine einheitliche Bewertung und Übersicht über die Unterschiedlichen Tools zu schaffen, unter intensiver Berücksichtigung der Anforderungen des Kunden.

5 Erstellung der Bewertungskriterien für Werkzeuge

Im Folgenden Kapitel werden die Bewertungskriterien des Kriterienkatalogs gesammelt und nach ihrer Relevanz gewichtet.

Die Bewertungskriterien teilen sich in zwei Arten von Kriterien auf, funktionale und nicht funktionale Kriterien. Die funktionalen Kriterien beschreiben die tatsächlichen Funktionen des Werkzeuges, wohingegen nicht funktionale Kriterien alle Aspekte beschreiben, die nicht direkt aus der Funktionalität heraus sichtbar sind, wie zum Beispiel die öffentlichen Bewertungen und der Reifegrad des Produktes.

Im Folgenden werden die Kriterien anhand der Tabellen 2 bis 14 beschrieben und gewichtet.

Funktionale Kriterien:

Funktionsumfang:

Name des Kriteriums	Beschreibung	Gewichtung
Product-Backlog	Wie sind die Möglichkeiten des Tools, ein Product-Backlog darzustellen?	7
Sprints	Wie werden Sprints dargestellt? Wie übersichtlich sind die Sprints?	6
Taskboard	Gibt es ein Taskboard? Wie übersichtlich ist das Taskboard?	5
Releases	Wie werden Releases dargestellt? Welche Informationen beinhalten diese?	5
User Stories	Wie werden User Stories erstellt? Wie übersichtlich sind diese?	7
Defects	Gibt es die Möglichkeit Defects anzulegen?	4
Tasks	Wie werden Tasks angelegt?	6
Kanban Page	Gibt es eine Kanban Page? Wie wird diese dargestellt?	5

Kanban Cards	Kann das Werkzeug Kanban Cards erzeugen? Wie sehen diese aus?	5
WIP/Spalten Limits	Gibt es bestimmte Spaltenlimits an die man sich halten muss? Gibt es eine „Work in progress“ Spalte?	1

Tabelle 2 Kriterien Funktionsumfang

Benutzeroberfläche:

Name des Kriteriums	Beschreibung	Gewichtung
Übersichtlichkeit	Wie übersichtlich ist das Werkzeug? Sind alle Funktionen einfach zu finden?	6
Einstiegsfreundlichkeit	Wie einstiegsfreundlich ist das Werkzeug? Wie schnell kann man sich in das Tool einarbeiten? Gibt es Möglichkeiten sich schneller einzuarbeiten?	7

Tabelle 3 Kritrien Benutzeroberfläche

Oberflächensteuerung:

Name des Kriteriums	Beschreibung	Gewichtung
Tiefe des Benutzerbaumes	Wie tief geht die Navigation? Wie vertieft liegen bestimmte Funktionen?	8
Zusammenspiel zwischen einzelnen Schritten	Wie gut kommunizieren die unterschiedlichen Schritte miteinander? Wie intuitiv ist dieses Zusammenspiel?	8
Administration	Wie gut können Personen und Informationen verwaltet werden? Wie viele Möglichkeiten gibt es zur Administration?	4

Tabelle 4 Kriterien Oberflächensteuerung

Auswertungsmöglichkeiten:

Name des Kriteriums	Beschreibung	Gewichtung
Burndown-Chart	Gibt es Burndown-Charts? Wie ausführlich und genau sind diese?	6

Velocity-Chart	Gibt es Velocity-Charts? Wie ausführlich und genau sind diese?	6

Tabelle 5 Kriterien Auswertungsmöglichkeiten

Schnittstellen:

Name des Kriteriums	Beschreibung	Gewichtung
Import/Export Dateien	Können Daten importiert oder exportiert werden?	4
Benutzerübergreifen-der Zugriff	Kann auf Informationen von unterschiedlichen Be-nutzern zugegriffen werden? Können Informationen privat gemacht werden?	5
Kommunikationsmög-lichkeiten Team	Bietet das Tool eine integrierte Möglichkeit für die Kommunikation zwischen Teammitgliedern? Gibt es eine E-Mail-Anbindung?	5

Tabelle 6 Kriterien Schnittstellen

<u>Nicht funktionale Kriterien:</u>

Markverbreitung:

Name des Kriteriums	Beschreibung	Gewichtung
Öffentliche Bewertun-gen	Gibt es öffentliche Bewertungen zu diesem Tool? Wie fallen diese Bewertungen aus?r	6
Bekannte Kunden	Gibt es bekannte Kunden, die das Tool verwenden?	3

Tabelle 7 Kriterien Marktverbreitung

Support:

Name des Kriteriums	Beschreibung	Gewichtung
Support Land	Aus welchem Land kommt der Support?	4
Support Art	Welche Art von Support wird angeboten? Telefon? Chat? E-Mail?	8

Tabelle 8 Kriterien Support

Dokumentation:

Name des Kriteriums	Beschreibung	Gewichtung
Art der Dokumentation	Um was für eine Dokumentation handelt es sich? Video-Tutorials, FAQ?	7
Sprache der Dokumentation	In welchen Sprachen ist die Dokumentation erhältlich?	6
Umfang der Dokumentation	Wie umfangreich ist die Dokumentation? Werden alle Gegebenheiten abgedeckt?	4

Tabelle 9 Kriterien Domkumentation

Kostenlose Version:

Name des Kriteriums	Beschreibung	Gewichtung
Lizenztyp	Welche Art von Lizenz hat das Tool? Ist es Open Source oder Freeware?	7
Kostenlose Version	Welchen Funktionsumfang hat die kostenlose Version des Tools?	7
Erweiterbarkeit	Was wird bei einer kostenpflichtigen Version mitgeliefert? Was für Zusatzleistungen bekommt man bei der kostenpflichtigen Version?	6
Upgrade Kosten	Wieviel kostet das Upgrade? Einmalige Kosten? Fixe Kosten? Variabel zur Benutzerzahl?	8

Tabelle 10 Kriterien kostenlose Version

Entwicklung:

Name des Kriteriums	Beschreibung	Gewichtung
Reifegrad	Wie ausgereift ist das Tool? Beta-Version? Fertiggestellt?	4
Update Stand (Release)	Wie oft wird das Tool upgedatet? Auf welcher Version befindet sich das Tool? Wie lange ist das letzte Update her?	4
Anzahl Entwickler	Wie viele Entwickler Arbeiten an dem Tool?	5

	Wie viele Entwickler sind in der Firma, die das Tool vertreibt?	

Tabelle 11 Kriterien Entwicklung

Nutzbarkeit:

Name des Kriteriums	Beschreibung	Gewichtung
Sprache	In welcher Sprache ist das Programm?	7
Art	Was für eine Art von Programm ist es? Webanwendung? Clientprogramm?	7
Umfang Installation	Wie umfangreich ist die Installation um das Programm im Unternehmen zu implementieren?	8
Voraussetzungen Einstieg	Was für Voraussetzungen gibt es, das Tool verwenden zu können? Was muss der Benutzer wissen?	7

Tabelle 12 Kriterien Nutzbarkeit

Methodenumfang:

Name des Kriteriums	Beschreibung	Gewichtung
Fähigkeit Scrum	Wie gut implementiert das Tool Scrum?	9
Fähigkeit Kanban	Wie gut implementiert das Tool Kanban?	9

Tabelle 13 Methodenumfang

Open Source Kriterien:

Name des Kriteriums	Beschreibung	Gewichtung
Lizenztyp	Welche Open Source-Lizenz besitzt das Tool?	5
Größe Community	Wie groß ist die Community hinter diesem Tool? Wie viele freie Entwickler tragen zu diesem Tool bei?	6

Tabelle 14 Kriterien Open Source

6 Marktübersicht über vorhandene Open Source Produkte

Damit man herausfinden kann, welches Werkzeug für agile Softwareentwicklung den zuvor definierten Kriterien am besten entspricht, muss man zuerst eine Übersicht darüber bekommen wie viele und welche Werkzeuge es überhaupt auf dem Markt gibt. Hierzu wird eine Marktanalyse durchgeführt, die bereits im Theorieteil 2.1 beschrieben wurde.

Die Grundlage besteht darin, sich eine Marktübersicht zu beschaffen. Der Markt wird nach Open Source und kostenlosen Werkzeugen durchsucht, die die im weiteren Verlauf erläuterten K.O.-Kriterien erfüllen.

Es ist essentiell, als Basis einen Überblick über alle vorhandenen kostenlosen Werkzeuge zu erhalten. Für diesen Zweck, wird eine Internetrecherche durchgeführt. Aus der Aufgabenstellung des Auftraggebers konnte die Internetseite „userstories.com" entnommen werden. Auf dieser Seite wurden bereits mehr als 60 Open Source Tools aufgelistet. Die Internetseite bietet viele Informationen zu den Werkzeugen der agilen Softwareentwicklung. Diese Webseite wurde für den Umfang des Projektes als ausreichend befunden und dient als Grundlage für die Marktübersicht. Ebenfalls wurde im Internet außerhalb von „userstories.com" gesucht.

Die zuvor genannte Webseite bietet die Möglichkeit an, die angezeigten Werkzeuge nach ihrer angegebenen Lizenz zu filtern. Die Differenz äußert sich demnach, ob die Werkzeuge eine kostenlose Version besitzen oder nur über kommerzielle Versionen verfügen. Nachdem die angezeigten Werkzeuge nach kostenlosen Versionen gefiltert wurden, verblieben noch 56 unterschiedliche Werkzeuge. Diese werden dann mit Hilfe von K.O.-Kriterien überprüft, ob sie sich grundsätzlich für eine genauere Betrachtung eignen. Die K.O.-Kriterien wurden aus den im Kapitel 5 erfassten Kriterien ausgewählt. Diese Kriterien bilden sich aus den wichtigsten Kriterien des Kriterienkatalogs. Eine Gewichtung wird in diesem Schritt nicht vorgenommen, die Erfüllung des Kriteriums reicht aus um bereits vorab die besten Tools herauszuarbeiten. Folgende Kriterien müssen erfüllt werden um später den weiteren Kriterien im Katalog unterzogen zu werden.

Die KO-Kriterien lauten wie folgt:

- Es wird überprüft, ob das Werkzeug in den letzten zwei Jahren upgedated wurde, was darauf hinweisen kann, ob ein Werkzeug vernachlässigt wurde.
- Die Homepage muss entweder auf Englisch oder auf Deutsch sein, damit man die Informationen auf der Seite verstehen kann.
- Das Werkzeug muss eine kostenlose Version besitzen, die sich nicht zu stark von der kommerziellen Version unterscheidet, was bedeutet dass sich die grundsätzliche Funktionalität nicht zu sehr unterscheiden soll.
  Ein akzeptabler Rahmen an Differenz ist dem Umfang zugerechnet, wie viele Benutzer das Werkzeug benutzen dürfen, ob ein Support vorhanden ist und wie viele Projekte verwaltet werden dürfen. Auch sind kleine Unterschiede in der Funktionalität vertretbar, solange diese das Werkzeug nicht zu stark einschränken.
  Werkzeuge mit „30 Tage-Probezeiten“ wurden hierbei nicht betrachtet.
- Das Werkzeug soll in seinem Reifegrad über die Beta-Phase herausgekommen sein und soll als volle Version veröffentlicht sein. Dies verhindert, dass man mit noch nicht ausgereifter Software arbeitet, die sich noch stark ändern kann oder auch Fehler aufweisen kann.
- Ebenfalls soll die Homepage des Werkzeuges erreichbar sein, um zu zeigen, dass die Entwickler des Werkzeuges Support anbieten können.

Die fast 60 Werkzeuge wurden mithilfe der KO-Kriterien und einer ersten Betrachtung nach den übrigen Kriterien auf fünf Werkzeuge eingegrenzt.

Diese fünf Werkzeuge werden im folgenden Kapitel nach dem zuvor definierten Bewertungskatalog untereinander verglichen.

7 Anwendung der Bewertungskriterien auf die Open Source Tools

Im Folgenden Kapitel werden die erarbeiteten Kriterien auf die zuvor ausgewählten Tools angewendet um diese zu vergleichen und zu bewerten. Das Ziel ist es, später das beste oder die besten Open Source Tools darzustellen und genauer zu beschreiben.

7.1 Eintragung der Open Source Tools in den Kriterienkatalog

Nachdem die Tools, die verglichen werden sollen, ausgewählt wurden, geht es daran diese nach einer objektiven und umfangreichen Weise zu vergleichen und zu bewerten. Hierfür ist ein Kriterienkatalog die beste Möglichkeit.

In den Kriterienkatalog werden die Werkzeuge mithilfe der in Kapitel 5 definierten und gewichteten Kriterien bewertet. Die Kriterien werden danach bewertet wie gut sie von den unterschiedlichen Tools erfüllt werden. Die Tools werden mit einem Wert von eins bis zehn bewertet, der mit der Gewichtung multipliziert wird. Am Ende, wenn alle Tools verglichen wurden, zeigt sich wie gut die einzelnen Tools im Vergleich zu den anderen Werkzeugen sind.

Eingetragen wurden die Tools:

- **Agilefant:**
  - Agilefant ist ein webbasiertes Open Source-Tool, das durch seine nicht funktionalen Kriterien heraus sticht. Hat den 2. Platz belegt und wird aus diesem Grund im Kapitel 7.2.2 genauer beschrieben. Das Tool erreichte 1768 Punkte.
- **iceScrum:**
  - Ice Scrum ist ein webbasiertes Open Source-Tool, das durch seine gute Funktionalität im Fokus stand. Das Tool erreichte 1755 Punkte.
- **Agility Ltd.:**
  - Agility Ltd. Ist ein webbasiertes Freeware-Tool. Das Tool kam durch sein simples Auftreten in die engere Wahl. Das Tool erreichte 1181 Punkte.
- **Kanboard:**
  - Kanboard ist ein webbasiertes Open-Source Tool, das auf Kanban spezialisiert ist. Durch diese Spezialisierung kam das Werkzeug in die engere Auswahl. Kanboard erreichte eine Gesamtpunktzahl von 1452 Punkten.

- **Tuleap:**
    - Tuleap ist ein Open-Source Web-Tool, das zur Unterstützung des Softwareentwicklungsprozesses entwickelt wurde. Das Tool unterstützt sowohl Scrum als auch Kanban. Tuleap belegte den ersten Platz und wird wie das Tool Agilefant im Kapitel 7.2.1 genauer beschrieben. Das Tool erreichte eine Gesamtpunktzahl von 1959 Punkten.

Der aufgefüllte Kriterienkatalog wird im Anhang 1 grafisch dargestellt, da die detaillierte Darstellung den Umfang dieses Kapitels übersteigen würde.

7.2 Kurzprofile der bestbewertesten Tools

Im Folgenden werden die zwei Produkte, die die höchste Punktzahl im Kriterienkatalog erzielten, beschrieben. Das Produkt Tuleap welches 1959 Punkte im Kriterienkatalog erreichte und Agilefant welcher 2450 Punkte erreichte.

7.2.1 Tuleap

Im Folgenden wird das Tool Tuleap beschrieben, da es im Kriterienkatalog positiv hervorgestochen ist.

Tuleap ist ein Open-Source Tool aus Frankreich von der Firma „enalean“, das auf Scrum ausgelegt ist, aber auch für Kanban und das Wasserfallmodell verwendet werden kann. Tuleap ist eine Server Anwendung, die zur Unterstützung der Softwareentwicklung eingesetzt werden kann.

Das Tool wurde durchgehend positiv bewertet und es wird angegeben, dass mehrere bekannte Unternehmen wie zum Beispiel Renault, Toyota und Delphi dieses Tool verwenden.

Tuleap ist voll ausgereift und wird regelmäßig geupdated. Zurzeit befindet sich das Tool in der Version 7.99.15.

Das Tool ist in den Sprachen Englisch und Französisch verwendbar.

Es wird ebenfalls eine sehr detaillierte Dokumentation angeboten, die mit vielen Grafiken und Videos gefüllt ist und einen sehr guten Überblick über die Möglichkeiten und Funktionen des Tools gibt. Die Dokumentation ist ebenfalls in Englisch und in Französisch erhältlich.

Der Einstieg ist sehr einfach und intuitiv. Man bekommt bereits zu Anfang einen Überblick über den Umfang des Tools (Abb. 5) und kann sofort anfangen, damit zu arbeiten.



Abb. 5: Einstiegsseite Tuleap

Für den Einstieg benötigt man lediglich Scrum Grundwissen um mit dem Tool anzufangen zu arbeiten.

Im Falle von Problemen oder bei Fragen gibt es mehrere Möglichkeiten Hilfe zu bekommen. Zum einen werden viele Fragen in der Dokumentation beantwortet, zum anderen kann man sich auf mehreren Wegen mit verschiedenen freien Entwickler und Entwickler der Firma „enalean" in Kontakt setzen. Es gibt die Möglichkeit, sich über ein Entwicklerforum helfen zu lassen, über ein Chatfenster kann man mit Entwicklern von „enalean" sprechen und auf der Homepage finden sich Telefonnummer und mehrere Emailadressen und –verteiler an die man sich mit Fragen wenden kann.

Tuleap ist ein Open-Source Tool unter der GNU GPL Lizenz, was bedeutet das das Tool kostenlos verwendet und ohne Einschränkungen verändert werden kann, so lange das Tool nicht als Eigenwerk ausgegeben wird. Das Tool hat eine große Community mit ungefähr 500 Entwicklern, die stetig an der Software weiterarbeiten.

Die Funktionalität kann nicht erweitert werden, was bedeutet dass die Open-Source Version den vollen Umfang der Funktionalität hat. Es kann jedoch zu einer „Enterprise"-Version erweitert werden, in der ein erweiterter professioneller Support bereit steht, so wie automatische Updates und Beratung bei Einsatz, Implementierung und Wartung. Der Preis dieser „Enterprise"-Version ist relativ zur Größe des Einsatzgebietes der Software und kann nur auf Nachfrage mit spezifischen Daten gegeben werden.

Das Werkzeug besitzt alles an Funktionalität was man zur Agilen Softwareentwicklung nach Scrum benötigt. Das Tool besitzt eine Product Backlog in dem alle noch nicht in Sprints bearbeiteten User Stories angezeigt werden. Die User Stories können einfach und schnell erstellt werden und ihnen kann vieles an Informationen zugegeben werden (Abb. 6).

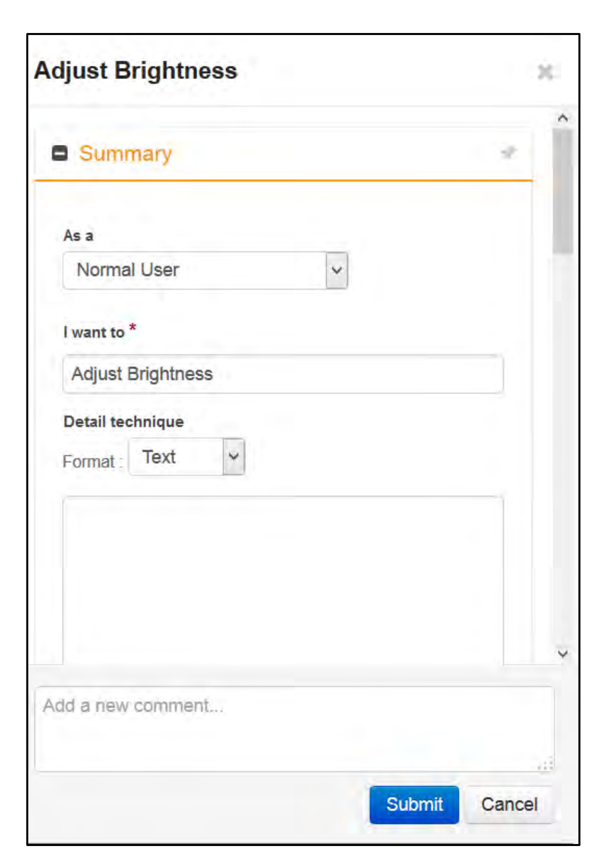


Abb. 6: Erstellung von User Story bei Tuleap

Sprints werden im Voraus geplant und werden durch ein Burndown Chart und ein Taskboard visualisiert. Die Tasks werden in der Taskliste erstellt und können dann zu bestimmten User Stories hinzugefügt werden. Das Werkzeug hat auch die Möglichkeit Defects zu erfassen.

Der Navigationsbaum, über die der Benutzer an seine unterschiedlichen Funktionen kommt ist klar und einfach strukturiert und geht nicht zu weit in die Tiefe. Dem Benutzer ist zu jedem Zeitpunkt klar, an welcher Stelle er sich befindet und welche Funktionen er dort verwenden kann oder soll.

Die Administration des Tools ist sehr ausführlich und man kann jeden notwendigen Punkt, der verwaltet gehört, verwaltet werden.

Die Auswertungsmöglichkeiten des Tools, wie die Burndown-Charts und Velocity-Charts, sind stark beschränkt, da das Tool mehr Wert auf das Unterstützen der Entwicklung legt, als auf das Auswerten der Statistiken.

Das Tool Tuleap hat eine Gesamtwertung von 1959 Punkten aus 2450 Punkten erzielt und hat somit den Platz 1 im Kriterienkatalog belegt. Das Tool sticht vor allem durch seine Funktionen und die Präsentation derer hervor. Das Tool ist einfach einzusetzen, sehr ausführlich und ist auf die freie Art der agilen Softwareentwicklung zugeschnitten. Das Tool kann sowohl für Scrum als auch für Kanban verwendet werden und ist somit eine gute Allgemeinlösung.

7.2.2 Agilefant

Im Folgenden wird das Tool Agilefant genauer beschrieben, da sich dieses im Kriterienkatalog durch seine Eigenschaften hervorgehoben hat. Agilefant ist eine Webanwendung aus Finnland zur Unterstützung von Agilen Projekten mit dem Schwerpunkt auf Scrum-Projekten.

Die Bewertungen des Tools sind durchgehend positiv und werden durch bekannte Kunden wie Yahoo! und ASP.NET bestätigt. Das Tool wird kontinuierlich aktualisiert und befindet sich inzwischen in der Version 3.5.2. Die verwendete Sprache des Tools ist Englisch.

Die Dokumentation ist durch ihren Umfang sehr hilfreich und wird durch Tutorials unterstützt. Als Einstieg wird ein Beispielprojekt bereitgestellt, um das Tool besser kennenzulernen. Die Entwickler des Tools bieten einen sehr umfangreichen Support über ein Chatfenster, User Guides und ein FAQ. Als Einstiegskriterium sind lediglich Grundkenntnisse in Scrum erforderlich.

Agilefant steht als Open-Source Tool zur Verfügung zum Hosten im eigenen Unternehmen, oder kann direkt als Cloudbetrieb verwendet werden gegen einen Endpreis. Agilefant wird unter der GNU GPL- Lizenz verbreitet. Agilefant besitzt eine Community von bis zu 800 Mitgliedern, welche aktiv im Forum beteiligt sind. Die cloudbasierte Version des Tools kann für bis zu fünf User kostenlos verwendet werden, ein Upgrade für 20 $ pro weiteren User und Monat kann dazu erworben werden.

Als Tool mit Schwerpunkt auf Scrum bietet Agilefant ein übersichtliches Product-Backlog, eine übersichtliche Darstellung aller Sprints mit Darstellung des Vollständigkeitsstands, ein Taskboard und leicht erstellbare User Stories. Die Releases sind ebenfalls vorhanden könnten jedoch klarer hervorgehoben werden. Funktionen für Kanban stehen nicht zur Verfügung.

Agilefant ist sehr übersichtlich, bereits die Einstiegsseite (Abb. 7) bietet alle nötigen Links auf einen Blick, wie die Dashboards, Reporting, und das Backlog. Über ein paar Knopfdrücke können jederzeit neue Produkte, Projekte, Iterationen (Sprints), Storys sowie Teams und User neu angelegt werden.

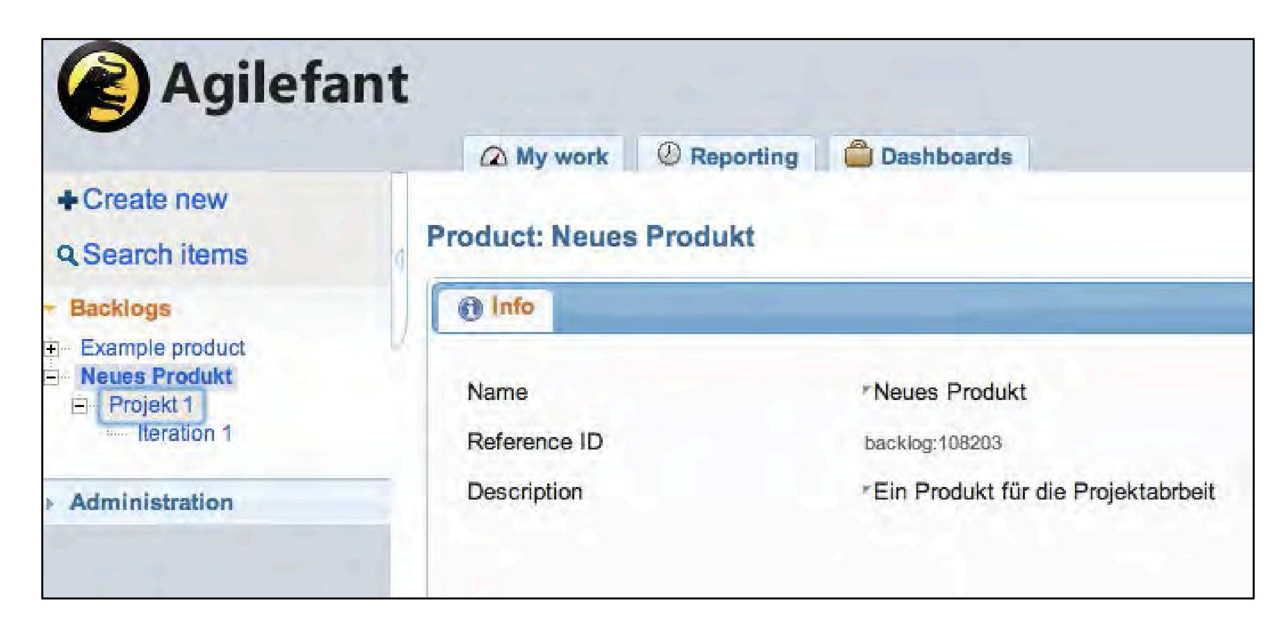


Abb.7: Darstellung Benutzerbaum

Die Tiefe des Benutzerbaums ist klar strukturiert, wird ein Projekt jedoch zu umfangreich können zu viele Verzweigungen entstehen. Das Tool bietet zudem eine übersichtliche und nützliche Administration, in welcher neue User und Teams angelegt werden können und die Auslastung des Teams eingetragen und verfolgt werden kann.

Die Projektfortschritte der Projekte können im Tool sehr einfach eingetragen werden, z. B. bei der Fertigstellung einer Story. Diese Fortschritte werden durch viele Auswertungsmöglichkeiten grafisch dargestellt, z. B. durch ein Burndown und mehrere Diagramme (Abb. 8), wie in folgender Grafik dargestellt.

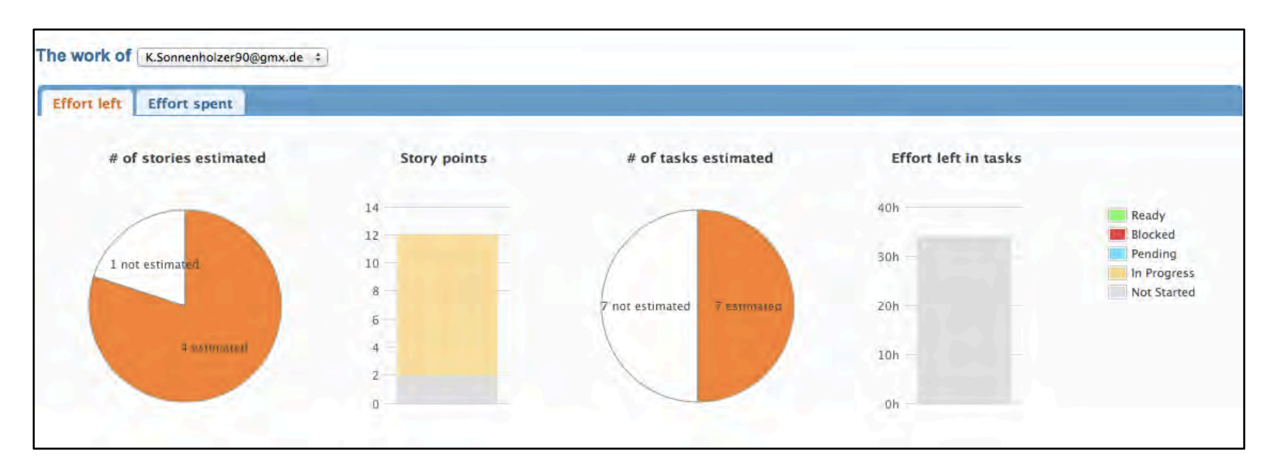


Abb. 8: Auswertungsmöglichkeiten bei Agilefant

Agilefant hat im Kriterienkatalog 1768 Punkte von 2450 Punkten erreicht und somit den Platz 2 belegt. Vor allem durch die Lizenzart OpenSource und den sehr umfangreichen Support stach dieses Tool besonders hervor. Für Scrum-Projekte ist dieses und Open-Source Tool eine sehr gute Wahl.

8 Fazit

Nach der Durchführung einer Marktanalyse im Internet wurden die vorhanden Freeware-Tools und Open-Source-Tools den fünf festgelegten KO-Kriterien unterzogen. Aus ca. 80 Tools wurden dann fünf potenziell geeignete Tools ausgewählt.

Durch die Erstellung eines Kriterienkatalogs, unter Berücksichtigung der Wünsche des Kunden, war es möglich die fünf ausgewählten Tools objektiv zu analysieren und miteinander zu vergleichen. So wurden zwei optimale Tools herausgearbeitet, welche den Anforderungen der Autoren und des Kunden entsprechen.

Mit 1959 Punkten stach das Tool Tuleap besonders hervor. Dieses Tool ist vor allem für Scrum zu empfehlen, weil dort die Funktionalität stark hervorsticht, gefolgt von Agilefant mit 1768 Punkten, welches durch die vielen Support Möglichkeiten begeisterte.

An dieser Stelle soll erwähnt werden, dass das Tool Kanboard das einzige Tool war, welches besonders die Kanban-Methode unterstützt und die KO-Kriterien erfüllte. Auf dem Markt verbreitete Tools zur Unterstützung von Agilen Methoden legen ihren Schwerpunkt häufig auf Scrum, Tools mit Schwerpunkt Kanban sind eher seltener. Das Tool Tuleap bietet jedoch einzelne Funktionsansätze wie ein Kanban Board.

9 Ausblick

Ziel dieser Arbeit war es, eine Marktanalyse zu den momentan am Markt befindlichen Tool zur Unterstützung „leichtgewichtiger SW-Entwicklung“ im Open Source Bereich durchzuführen und eine Bewertungsgrundlage zur Auswahl eines auf die Anforderungen des jeweiligen Unternehmens passenden Tools zu schaffen.

Durch die Durchführung einer Marktanalyse, K.O.-Kriterien und einem Kriterienkatalog konnten die Tools objektiv auf zwei eingegrenzt werden. Hieraus folgt, wie bereits im Fazit dargestellt, die Empfehlung für den Einsatz des Tools Tuleap oder Agilefant im Unternehmen.

Jedes Unternehmen und jedes Projekt hat jedoch individuelle Wünsche, Anforderungen und Kriterien an solch ein Tool. Es muss somit für jedes Projektteam entschieden werden welche agilen Methoden sie einsetzen, wie groß die Teams und Projekte sind, wie viele Funktionen sie im Tool benötigen und welche funktionalen und nicht funktionalen Anforderungen das später einzusetzende Tool benötigt. Unter zu Hilfenahme des in dieser Arbeit erstellen Kriterienkatalogs kann dann das optimalste Tool ermittelt werden.

In der heutigen globalen Welt in welcher nicht mehr unbedingt alle Projektmitglieder zusammen in einem Raum am Projekt arbeiten, ist ein Tool zur Unterstützung von Agilen Methoden zu empfehlen. Der Mehrwert den diese Tools unter der richtigen Anwendung einem Unternehmen bieten können ist sehr hoch. Zu diesem Mehrwert gehören die Übersichtlichkeit, administrativen Möglichkeiten, die uneingeschränkten Räumlichen Möglichkeiten in der Zeit von Homeoffice und Globalisierung und bessere Lesbarkeit. Durch die hohe Qualität und Vielzahl von Open-Source- und Freeware- Tools sind die entstehenden Kosten zudem sehr gering bis gar nicht vorhanden. Weshalb Open-Source-Tools sehr zu empfehlen sind und die Zukunft in den Unternehmen sind.

Anhang

Anhangsverzeichnis:

Anhang 1: Kriterienkatalog

Nutzwertanalyse von kostenlosen Tools für die leichtgewichtige Softwareentwicklung							
Kriterien		**Agilefant**			**IceScrum**		
Funktionale Anforderungen	**Gewichtung**	**Begründung**	**Bewertung**	**Score**	**Begründung**	**Bewertung**	**Score**
Funktionsumfang	51			258			288
Product-Backlog	7	+übersichtlich +Produktbaum aufklappbar -abkürzungen auf ersten Blick nicht offensichtlich	7	49	+übersichtlich +auf Backlog kann von überall her zugegriffen werden	8	56
Sprints	6	+alle zu einem Sprint gehörigen User-Stories auf einen Blick +Aufwände sichtbar dargestellt +Vervollständigungsstand sichtbar	9	54	+Sprints werden nacheinander erstellt und Aufgaben werden im Sprint hinzugefügt und bewertet +User Stories können aus dem Backlog in die Sprints gezogen werden	8	48
Taskboard	5	+Alle momentan offenen Tasks sichtbar +Umfang klar erkennbar	7	35	+Taskboard direkt im Sprint mit aktuellem Status +Aufgaben werden direkt in ihrer Zugehörigkeit erstellt	8	40
Releases	5	-Releases nicht klar erkennbar	3	15	+Releases klar erkennbar	8	40
User Stories	7	+leicht erstellbar +Umfang variabel	9	63	+leicht erstellbar +viele Beschreibungsmöglichkeiten	8	56
Defects	4	- nicht vorhanden	0	0	-nicht vorhanden	0	0
Tasks	6	+werden im Backlog erzeugt	7	42	+werden im Taskboard erstellt	8	48
Kanban Page	5	-nicht vorhanden	0	0	-nicht vorhanden	0	0
Kanban Cards	5	-nicht vorhanden	0	0	-nicht vorhanden	0	0
WIP/Spaltenlimits	1	-nicht vorhanden	0	0	-nicht vorhanden	0	0
Benutzeroberfläche	13			111			104
Übersichtlichkeit	6	+klar gegliedert	8	48	+sehr übersichtlich	8	48
Einstiegsfreundlichkeit	7	+leichter Einstieg durch Support und Tutorials +Beispielprojekte vorhanden	9	63	+Einstieg durch Tutorials Sehr einfach -User Guide ist teilweise nötig um Funktionen zu finden	8	56
Oberflächensteuerung	20			120			148
Tiefe des Benutzerbaumes	8	+ tiefe klar erkennbar -Funktionen teilweise zu tief	5	40	+sehr klare Tiefe +alle Funktionen auf hohen Ebenen	8	64
Zusammenspiel zw. einzelnen Schritten	8	+teilweise gut gegeben -in einzelnen Fällen zu unabhängig und unübersichtlich	5	40	+immer gegeben + Zusammenspiel immer klar	8	64
Administration	4	+Verwaltung aller Benutzer gut +Rechteverwaltung +Es kann vieles übersichtlich verwaltet werden	10	40	+Teammitglieder können einfach verwaltet werden	5	20
Auswertungsmöglichkeiten	12			108			108
Burndown-Chart	6	+Legende +alles erkennbar +farbliche Unterschiede	9	54	+Burndown vorhanden auf Dashboard und in Statistiken	9	54
Velocity Chart	6	+in Burndown integriert	9	54	+in Statistiken eingegliedert	9	54
Schnittstellen	14			86			82
Export / Import Excel-Dateien	4	+es können Dateien Expotiert werden	9	36	+Es können Dateien exportiert werde	8	32
Benutzerübergreifender Zugriff	5	+möglich	10	50	+möglich	10	50
Kommunikationsmöglichkeiten Team	5	-keine integrierte kommunikation	0	0	-keine integrierte Kommunikation	0	0
Total Funktionale Anforderungen	[illegible]			683			730

Nichtfunktionale Anforderungen							
Marktverbreitung	9			69			48
✓ Öffentliche Bewertungen	6	+öffentliche Bewertungen durchgehend positiv	8	48	+durchgehend positive Bewerungen	8	48
✓ Bekannte Kunden	3	+ mehrere bekannte Firmen (Bsp. Yahoo!, ASP	7	21	-keine genannt	0	0
Support	12			88			76
✓ Support Land	4	+Finnland	8	32	+Frankreich	7	28
✓ Support Art (Hotline oder Forum)	8	+Chat, User Guide, FAQ, Forum	7	56	+Email, Forum, Twitter, Facebook, Telefon	6	48
Dokumentation	21			170			154
✓ Art der Dokumentation	7	+Videos, Anleitungen, FAQ	8	56	+Anleitung -muss registriert sein um Dokumentation zu sehen	6	42
✓ Sprache der Dokumentation	6	+Englisch -Keine anderen Sprachen	7	42	+Englisch +Französisch	8	48
✓ Umfang der Dokumentation	8	+Sehr umfangreich	9	72	+Umfangreiche Dokumentationen	8	64
Kostenlose Version	28			248			209
✓ Lizenztyp	7	+Open-Source, wenn selbst gehostet	10	70	Open-Source für selbst hosten	10	70
✓ Umfang kostenlose Version	7	+voller Umfang für eigene Server	10	70	"einfache" Version	7	49
✓ Erweiterbarkeit	6	+erweiterbar durch Fremdhosten +Funktionalität nicht erweiterbar	10	60	Erweiterbar als Pro Version Erweiterbar durch fremdhosten	7	42
✓ Upgrade Kosten	8	+Preis abhängig von Useranzahl (20$ p.P)	6	48	Cloud Version abhängig von Benutzern Pro version 2000 Euro im Jahr	6	48
Entwicklung	13			126			118
✓ Reifegrad	4	+voll ausgereift	10	40	+voll augereift	9	36
✓ Update Stand (Release)	4	+kontinuierlich upgedated +Version 3.5.2	9	36	kontinuierlich geupdated Version R6#13.3	8	32
✓ Anzahl Entwickler	5	+durch Open-Source und Entwickler-Forum viele Entwickler	10	50	+durch Open Source Viele Entwickle	10	50
Nutzbarkeit	29			232			238
✓ Sprache	7	+Englisch -keine andere Sprache	7	49	+Englisch +Deutsch +Spanisch +Französisch	9	63
✓ Art (Browser oder Client-Anwendung)	7	+Browser-Anwendung	8	56	+Browser-Anwendung	8	56
✓ Umfang Installation	8	+Server-Daten für Apache/Tomcat-Server	8	64	+Serveranwendung	7	56
✓ Voraussetzungen für den Einstieg	7	+Internet wenn nicht selbst gehostet + Grundkenntnisse Scrum	9	63	+Notwendige Dateien +Grundkentnisse Scrum	9	63
Methodenumfang	12			48			72
✓ Fähigkeit Scrum	6	+Umfang gut gegeben +Für Scrum entwickelt	8	48	+Für Scrum entwickelt	8	48
✓ Fähigkeit Kanban	6	-nicht vorhanden	0	0	+Bestimmte Features vorhanden	4	24
Open Source Kriterien	11			104			110
✓ Lizenzart	5	+Open Source	10	50	GNU GPL	10	50
✓ Größe Community	6	+ ca. 800 Member, 50 durchschnittlich online +Aktives Forum	9	54	+1500 Mitglieder	10	60
Total Nichtfunktionale Anforderunge	135			1085			1025
Gesamtscore	2450			1768			1755
Platzierung				2			3

Nutzwertanalyse von kostenlosen Tools für die leichtgewichtige Softwareentwicklung							
Kriterien		Agility Ltd.			Kanboard		
Funktionale Anforderungen	Gewichtung	Begründung	Bewertung	Score	Begründung	Bewertung	Score
Funktionsumfang	51			272			247
Product-Backlog	7	+Das Backlog ist sehr übersichtlich	8	56	+sehr einfach gehalten	7	49
Sprints	6	+ Sprints sind übersichtlich	8	48	-nicht vorhanden	0	0
Taskboard	5	+Die Tasks können Sehr detailliert beschrieben werden	7	35	+Kanban Page	8	40
Releases	5	+klar übersichtlich	7	35	-nicht vorhanden	0	0
User Stories	7	+können sehr detailiert beschrieben werden	8	56	-nicht vorhanden	0	0
Defects	4	-nicht vorhanden	0	0	-nicht vorhanden	0	0
Tasks	6	+Die Tasks sind alle auf einer Seite übersichtlich dargestellt mit Verlinkung zu den Storys und dem aktuellen Stand	7	42	+können einfach erstellt werden	8	48
Kanban Page	5	-nicht vorhanden	0	0	+als Taskboard implementiert	10	50
Kanban Cards	5	-nicht vorhanden	0	0	+Als Tasks implementiert	10	50
WIP/Spaltenlimits	1	-nicht vorhanden	0	0	+Frei erstellbar +Kein Spaltenlimit	10	10
Benutzeroberfläche	13			105			117
Übersichtlichkeit	6	+ auf einen blick sind die wichtigsten Funktionen dargestellt	7	42	+sehr einfache Oberfläche	9	54
Einstiegsfreundlichkeit	7	+ Der Eintieg ist sehr einfach und klar + Das Tool ist selbsterklärend	9	63	+Einstieg sehr intuitiv	9	63
Oberflächensteuerung	20			152			144
Tiefe des Benutzerbaumes	8	+ Die Tiefe des Benutzerbaums ist logisch und einfach dargestellt	8	64	+kaum Tiefe, alles auf fast gleicher Eb	8	64
Zusammenspiel zw. einzelnen Schritten	8	+ Die Verbindung zwischen den einzelnen Funktionen ist logisch	8	64	+teilweise gut gegeben	7	56
Administration	4	+ Teams können einfach zusammengestellt werden + weitere Administationen sind nicht möglich	6	24	-Verwaltung nur sehr beschränkt	6	24
Auswertungsmöglichkeiten	12			48			0
Burndown-Chart	6	+ Schlicht und Übersichtlich +Legende	8	48	-nicht vorhanden	0	0
Velocity Chart	6	-nicht vorhanden	0	0	-nicht vorhanden	0	0
Schnittstellen	14			50			86
Export / Import Excel-Dateien	4	- nicht möglich	0	0	+möglich	9	36
Benutzerübergreifender Zugriff	5	+möglich	10	50	+möglich	10	50
Kommunikationsmöglichkeiten Team	5	-nicht vorhanden	0	0	-nicht vorhanden	0	0
Total Funktionale Anforderungen	110			627			594

Nichtfunktionale Anforderungen							
Marktverbreitung	9			0			72
Öffentliche Bewertungen	6	-nicht vorhanden	0	0	gute Bewertungen	8	48
Bekannte Kunden	3	-nicht vorhanden	0	0	nicht gegeben	8	24
Support	12			0			0
Support Land	4	-Support ist keiner vorhanden	0	0	-nicht vorhanden	0	0
Support Art (Hotline oder Forum)	8	-Support ist keiner vorhanden	0	0	-nicht vorhanden	0	0
Dokumentation	21			72			147
Art der Dokumentation	7	- kleine FAQ	2	14	+Dokumentation umfassend	7	49
Sprache der Dokumentation	6	+ Englisch - keine weitere Sprache vorhanden	7	42	+Englisch	7	42
Umfang der Dokumentation	8	- sehr gering	2	16	+umfangreich	7	56
Kostenlose Version	28			161			210
Lizenztyp	7	-Freeware, kein Open-Source	3	21	+Open Source	10	70
Umfang kostenlose Version	7	+voller Funktionsumfang	10	70	+voller Funktionsumfang	10	70
Erweiterbarkeit	6	-Es steht keine Erweiterbarkeit zur Verfügung	5	30	-Keine Erweiterungen	5	30
Upgrade Kosten	8	-s.O	5	40	-s.O	5	40
Entwicklung	13			40			40
Reifegrad	4	+ voll ausgereift	10	40	+voll ausgereift	10	40
Update Stand (Release)	4	- nicht ermittelbar	0	0	-nicht ermittelbar	0	0
Anzahl Entwickler	5	- nicht ermittelbar	0	0	-nicht ermittelbar	0	0
Nutzbarkeit	29			208			225
Sprache	7	+ Englisch	7	49	+Englisch	7	49
Art (Browser oder Client-Anwendung)	7	+ Browser Anwendung	8	56	+Browser-Anwendung	8	56
Umfang Installation	8	- keine Installationsdaten	5	40	+Serverdaten	8	64
Voraussetzungen für den Einstieg	7	+ geringe Grundkenntnisse Scrum	9	63	+Kanban Kenntnisse	8	56
Methodenumfang	12			48			78
Fähigkeit Scrum	6	+ Für Scrum entwickelt	8	48	-Scrum nur beschränkt möglich	4	24
Fähigkeit Kanban	6	- nicht vorhanden	0	0	+Auf Kanban entwickelt	9	54
Open Source Kriterien	11			25			86
Lizenzart	5	Freeware	5	25	GNU GPL-Lizenz	10	50
Größe Community	6	-Es existiert keine Community	0	0	-30 Mitglieder	6	36
Total Nichtfunktionale Anforderungen	[illegible]			**554**			**858**
Gesamtscore	2450			**1181**			**1452**
Platzierung				5			4

Nutzwertanalyse von kostenlosen Tools für die leichtgewichtige Softwareentwicklung				
Kriterien		**Tuleap**		
Funktionale Anforderungen	**Gewichtung**	**Begründung**	**Bewertung**	**Score**
Funktionsumfang	51			424
✓ Product-Backlog	7	+Backlog wird klar dargestellt, alle User Stories Auf einen Blick erkennbar +Art der Story erkennbar +Stories in Sprints automatisch herausgenommen	9	63
✓ Sprints	6	+Sprints sind klar dargestellt mit einem Burndown	9	54
✓ Taskboard	5	+Task werden sehr detailiert angezeigt +Status von Tasks wird klar dargestellt	10	50
✓ Releases	5	+Releases klar erkennbar	8	40
✓ User Stories	7	+einfach erstellbar +verbundenen Tasks und Defects erkennbar	9	63
✓ Defects	4	+Defects können angelegt werden mit unterschiedlicher Priorität	9	36
✓ Tasks	6	+Tasks werden in einer allgemeinen Taskliste erstellt und dann bestimmten User Stories ggf. zugewiesen	8	48
✓ Kanban Page	5	+ Im Taskboard integriert	7	35
✓ Kanban Cards	5	+ Als Tasks integriert	7	35
✓ WIP/Spaltenlimits	1	-nicht vorhanden	0	0
Benutzeroberfläche	13			116
✓ Übersichtlichkeit	6	+sehr Übersichtlich +Suchfunktion +Tracker für alle Projektinhalte	10	60
✓ Einstiegsfreundlichkeit	7	+Einstieg sehr leicht +Beispielprojekt im Internet vorhanden	8	56
Oberflächensteuerung	20			176
✓ Tiefe des Benutzerbaumes	8	+logisch aufgebaut +nicht zu tief mit vielen Weiterleitungen auf andere Strukturen	9	72
✓ Zusammenspiel zw. einzelnen Schritter	8	+ Verbindungen sind intuitiv + Alle notwendigen Schritte sind verbunden - Tasks können nicht direkt in den Stories erstellt werden	8	64
✓ Administration	4	+Sehr umfangreiche Personalverwaltung +Sehr gute Datenverwaltung	10	40
Auswertungsmöglichkeiten	12			54
✓ Burndown-Chart	6	+wird für jeden Sprint automatisch erstellt und aktualisiert	9	54
✓ Velocity Chart	6	-nicht vorhanden	0	0
Schnittstellen	14			140
✓ Export / Import Excel-Dateien	4	+Export bestimmter Daten möglich +Import möglich	10	40
✓ Benutzerübergreifender Zugriff	5	+möglich	10	50
✓ Kommunikationsmöglichkeiten Team	5	+Kommentar Möglichkeiten bei Items +Chat	10	50
Total Funktionale Anforderungen	110			910

Nichtfunktionale Anforderungen				
Marktverbreitung	9			72
Öffentliche Bewertungen	6	+ positive Bewertungen	8	48
Bekannte Kunden	3	+mehrere bekannte Unternehen z.B Renault, Toyota, Delphi	8	24
Support	12			84
Support Land	4	+Frankreich	7	28
Support Art (Hotline oder Forum)	8	+Email, Forum, Telefon, User Guide +Chat in Cloud Version	7	56
Dokumentation	21			177
Art der Dokumentation	7	+Sehr ausführliche Dokumentation +Viele Grafiken	7	49
Sprache der Dokumentation	6	+Englisch +Französisch	8	48
Umfang der Dokumentation	8	+Sehr umfangreiche Dokumentation +Wiki +Tutorials	10	80
Kostenlose Version	28			212
Lizenztyp	7	Open Source Für selbst hosten	10	70
Umfang kostenlose Version	7	Voller Funktionsumfang	10	70
Erweiterbarkeit	6	"Enterprise" Version mit erweitertem Support Automatische Updates	8	48
Upgrade Kosten	8	Preise nur auf Nachfrage	3	24
Entwicklung	13			102
Reifegrad	4	voll ausgereift	10	40
Update Stand (Release)	4	Kontinuierliche Updates Version 7.8.2	8	32
Anzahl Entwickler	5	+Wartung regelmäßig bei Enterprise Version	6	30
Nutzbarkeit	29			232
Sprache	7	+Englisch +Französisch	8	56
Art (Browser oder Client-Anwendung)	7	+Browser-Anwendung	8	56
Umfang Installation	8	+Server Daten	8	64
Voraussetzungen für den Einstieg	7	+Grundwissen Scrum/Kanban	8	56
Methodenumfang	12			72
Fähigkeit Scrum	6	+Für Scrum entwickelt	8	48
Fähigkeit Kanban	6	+teilweise Funktionen integriert	4	24
Open Source Kriterien	11			98
Lizenzart	5	GNU GPL-Lizenz	10	50
Größe Community	6	mehr als 500 Mitglieder	8	48
Total Nichtfunktionale Anforderungen	135			1049
Gesamtscore	2450			1959
Platzierung				1

Quellenverzeichnisse

Literaturverzeichnis

.Versicherung (2011): Kanban, in: Evolutionäres Change Management für IT Organisationen, Mai, München, S. 1-12

Büsch, M. (2013): Praxishandbuch Strategischer Einkauf, Methoden, Verfahren, Arbeitsblätter für professionelles Beschaffungsmanagement, 3. Auflage, Wiesbaden: Springer Fachmedien

Deutsches Institut für Interne Revision e.V. (2011): Revision der Beschaffung, Prüfungsfragen für die Praxis, 4. Auflage, Berlin: Erich Schmidt Verlag GmbH und Co. KG

Dickmann, P. (2007): Schlanker Materialfluss, Mit Lean Production, Kanban und Innovationen, Heidelberg: Springer Berlin Heidelberg

Fox, D./Horster, P. (2006): Open Source, in: Datenschutz und Datensicherheit, 30. Jg., Nr. 6, Wiesbaden: Springer-Verlag, S. 371

Gloger, B. (2010): Scrum, in: Informatik-Spektrum, 2010, 33. Jg., Nr. 2, S. 195-200

Hanser, E. (2010): Agile Prozesse: Von XP über Scrum bis MAP, Berlin/ Heidelberg: Springer-Verlag

Haun, M. (2014): Cognitive Computing, Berlin/Heidelberg: Springer-Verlag

Kleuker, S. (2013): Grundkurs Software-Engineering mit UML, Wiesbaden: Springer Fachmedien

Schuh G./Schmidt C. (2014): Produktionsmanagement, Handbuch Produktion und Management 5, 2. Auflage, Berlin und Heidelberg: Springer Verlag

Internet- und Intranetquellen

3m5 (2011): 3m5., https://www.3m5.de/typo3temp/pics/92f8313087.jpg, Abruf: 10.12.2014

Beck, K./Beedle, M./van Bennekum, A. (2001): Manifest für Agile Softwareentwicklung, http://agilemanifesto.org/iso/de/, Abruf 16.09.2014

Dickmann, P./Dickmann E. (2009): Kanban-System oder Pull-Steuerung - was ist das?, http://www.lepros.de/kanban-definition.php?id=21, Abruf: 20.11.2014

Duden (o. J.): Wörterbuch, http://www.duden.de/rechtschreibung/agil, Abruf: 10.09.2014

it-agile (o. J.): Scrum in 5 minuten, http://www.it-agile.de/wissen/videos/scrum-in-5-minuten/, Abruf: 22.11.2014

it-agile (o. J.): Kanban, http://www.it-agile.de/wissen/methoden/kanban/, Abruf: 15.01.2014

IT Wissen (2013): V-Modell, http://www.itwissen.info/definition/lexikon/Vorgehensmodell.html, Abruf: 18.11.2014

ifrOSS (o.J.): „Welches sind die wichtigsten Open Source Lizenen und welchem Lizenztyp gehören sie an?", http://www.ifross.org/welches-sind-wichtigsten-open-source-lizenzen-und-welchem-lizenztyp-gehoeren-sie, Abruf: 22.01.2015

Klipp, P. (2010): Cumulative Flow Diagramm, http://paulklipp.com/images/Interpreting_a_Cumulative_Flow_Diagram.jpg, Abruf: 20.01.2015

Kuhrmann, M. (2013): Enzyklopädie der Wirtschaftsinformatik, http://www.enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/lexikon/is-management/Systementwicklung/Vorgehensmodell/Agile-Vorgehensmodelle/index.html, Abruf: 10.09.2014

LEAN Managementmethoden (2015): Lean Einführung & Reifegrad Ihres Unternehmens http://lean-managementmethode.de/lean/einfuehrung_reifegrad_lean/, Abruf: 12.01.2015

mindmeister (2013): Unterschied zwischen Scrum und Kanban, http://www.mindmeister.com/de/99946808/unterschiede-zwischen-scrum-kanban, Abruf: 20.12.2014

Scrum Kompakt (2014): Einführung in Scrum, http://www.scrum-kompakt.de/einfuehrung-in-scrum/fortschrittskontrolle-im-sprint/, Abruf: 29.11.2014

Sutherland, J./Schwaber, K. (2014): Scrum Guides, http://www.scrumguides.org/, Abruf: 22.11.2014

Wübbenhorst, K. (o. J.): Marktanalyse, http://wirtschaftslexikon.gabler.de/Archiv/378/marktanalyse-v8.html, Abruf: 26. November 2014

WinfWiki (2014): Vergleich zwischen Scrum und Kanban, http://winfwiki.wi-fom.de/index.php/Vergleich_zwischen_SCRUM_und_Kanban, Abruf: 26.11.2014

Sonstige Literatur

Haberl, P./Spillner, A./Vosseberg, K./ (2012): Umfrage Softwaretest in der Praxis, Heidelberg: dpunkt.verlag GmbH



Testszenarien für NoSQL-Datenbanksysteme/-dienste aus der Cloud

Schriftliche Ausarbeitung
im Rahmen der Lehrveranstaltung „Integrationsseminar“
für das Kompetenzzentrum Open Source (KOS)

Vorgelegt von

Johannes Haug, Konstantin Kübler,
Dennis Rothfuss

am 23.01.2015

Fakultät Wirtschaft
Studiengang Wirtschaftsinformatik
WWI2012V

Inhaltsverzeichnis

Abkürzungsverzeichnis

AWS — **A**mazon **W**eb **S**ervices

CQL — **C**ontinuous **Q**uery **L**anguage

CSV — **C**omma-**s**eparated **v**alues

DBMS — **D**aten**b**ank**m**anagement**s**ystem

HTML — **H**yper**t**ext **M**arkup **L**anguage

IaaS — **I**nfrastructure-**a**s-**a**-**S**ervice

IAM — **I**dentity and **A**ccess **M**anagement

JSON — **J**ava**S**cript **O**bject **N**otation

NoSQL — **N**ot **o**nly **S**tructured **Q**uery **L**anguage

OLAP — **O**n**l**ine **A**nalytical **P**rocessing

PaaS — **P**latform-**a**s-**a**-**S**ervice

PDF — **P**ortable **D**ocument **F**ormat

RDBMS — **r**elationales **D**aten**b**ank**m**anagement**s**ystem

SDK — **S**oftware **D**evelopment **K**it

SLA — **S**ervice-**L**evel-**A**greement

SSH — **S**ecure **Sh**ell

VM — **v**irtual **m**achine

XLS — Microsoft Excel-Datei

XML — E**x**tensible **M**arkup **L**anguage

ZIP — Format für komprimierte Dateien (von englisch: **zip**per)

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

In Social-Networks, auf Blogs oder anderen Plattformen werden täglich neue Inhalte mit der Welt geteilt. Und auch Unternehmen sammeln unzählige Daten, um beispielsweise ihre Geschäftsstrategien an veränderte Marktanforderungen anpassen zu können. Das ständig wachsende Datenvolumen stellt Datenbanken vor neue Herausforderungen. Oft kommen klassische relationale Datenbanken dabei an ihre Grenzen, weil durch eine Zunahme der Daten die Server ausgelastet sind. Das ist einer der Gründe, weshalb klassische Datenbanklösungen seit einigen Jahren immer stärkere Konkurrenz durch so genannte NoSQL-Systeme bekommen. Diese können mit dem ständigen Datenwachstum besser und oft vor allem kostengünstiger umgehen.

Ein Thema, was sich in der Bevölkerung weit größerer Beachtung als die Diskussion um Datenbanksysteme erfreut, ist der Gedanke an Cloud-Computing. Die Sammlung von Dokumenten, Bildern und anderen Daten hat den Reiz, dass eben diese Informationen prinzipiell an jedem Ort und von jedem Endgerät aus abrufbar sind. Nicht nur Privatpersonen haben die Vorteile dieser Technologie für sich entdeckt. Auch Unternehmen beginnen zunehmend damit, Daten auf Server umzusiedeln, um sie in einer Cloud nutzbar zu machen.

Diese wissenschaftliche Arbeit beschäftigt sich mit der Frage, wie Unternehmensdaten in der Cloud nutzbar gemacht werden können. Hierbei soll vor allem der Umgang mit Daten im Vordergrund stehen, die für die Versicherungsbranche typisch sind. Von der Betrachtung klassischer Datenbankmodelle wird dabei abgesehen. Stattdessen steht im Vordergrund, welche Funktionalitäten NoSQL-Lösungen in der Cloud anbieten.

In den folgenden Kapiteln wird, nach einem kurzen theoretischen Überblick, eine Auswahl von zwei NoSQL-Datenbanken und zwei Cloud-Providern getroffen. Die gewählten Datenbanken werden anschließend auf je einer der unterschiedlichen Clouds implementiert. Abschließend werden die Systeme mit Testdaten befüllt und auf ihre Funktionen und Performance getestet. Die Bereitstellung der Testdatenbank ist ebenfalls Teil dieser wissenschaftlichen Arbeit. Hierbei ist geplant, die Datensätze aus Open-Data-Beständen abzuziehen. Im Zuge dessen erfolgt auch eine grobe Analyse der Auswahl an offenen Daten im Internet.

Ziel der Arbeit ist ein Bild davon zu vermitteln, wie weit der technische Reifeprozess von NoSQL-Datenbanken in der Cloud bereits vorangeschritten ist. Dabei soll insbesondere ersichtlich werden, ob NoSQL-Systeme eine Alternative zu klassischen Datenbanksystemen in Unternehmen darstellen.

2 Allgemeines

2.1 Begriffserklärung Open Data

Unter Open Data oder übersetzt „offenen Daten" versteht man Daten, die für jede Person frei zugänglich sind und beliebig bearbeitet oder verbreitet werden dürfen. Dabei spielt es keine Rolle für welchen Zweck die Daten verwendet werden.[1] Zusammenfassend lassen sich offene Daten wie folgt definieren: „Daten sind dann offen, wenn es keine rechtlichen, technischen oder sonstigen Kontrollmechanismen gibt, die den Zugang, die Weiterverarbeitung und die Weiterverbreitung dieser Daten einschränken. Der Zugang, die Weiterverarbeitung und die Weiterverbreitung soll jedermann und zu jeglichem Zweck, auch kommerziellem, ohne Einschränkungen und Diskriminierung und ohne Zahlung von Gebühren möglich sein".[2]

Personenbezogene Daten dürfen in der Regel nicht zugänglich gemacht werden. Ausnahmen gibt es bei anonymisierten Datensätzen. Im öffentlichen Sektor lassen sich zudem keine Daten finden, deren Veröffentlichung ein Sicherheitsrisiko darstellen würde. Erkennen lassen sich offene Datenbestände im Internet daran, dass sie heruntergeladen werden können, beziehungsweise die Abfrage über eine Schnittstelle möglich ist.[3]

Oft handelt es sich bei offenen Datenbeständen um Rohdaten, wie zum Beispiel Klima- oder Geodaten.[4] Eine Auflistung verschiedener Datentypen in offenen Beständen kann *Abb. 1* entnommen werden.

[1] Vgl. Dietrich, D. (2011)
[2] Ebenda
[3] Vgl. Matzat, L. (2011)
[4] Vgl. Dietrich, D. (2011)



Abb. 1: Typen offener Daten[5]

Inzwischen wird der Begriff Open Data häufig synonym für Open Government Data verwendet. Diese sind Daten die von Länderbehörden veröffentlicht werden, so zum Beispiel Umwelt-, Verkehrs-, und Haushaltsdaten aber auch Statistiken, Protokolle oder Gesetze und Verordnungen. Grundsätzlich schließt der Begriff Open Data aber auch Daten aus Wissenschaft und Forschung ein.[6]

Es gibt eine Vielzahl von Empfehlungen, wie Regierungs-Datenbestände der Allgemeinheit zugänglich gemacht werden sollen. Als Standard gelten inzwischen die 2010 veröffentlichten „Ten Principles for Opening Up Government Information“ (übersetzt: zehn Prinzipien zum Öffnen von Regierungsinformationen).[7] Diese zehn Prinzipien lauten wie folgt:[8]

1) Completness
2) Primacy
3) Timeliness
4) Ease of Physical and Electronic Access
5) Machine readability
6) Non-discrimination

[5] Enthalten in: OKF DE (o.J.)
[6] Vgl. Dietrich, D. (2011)
[7] Vgl. Ebenda
[8] Vgl. Sunlight Foundation (2010)

7) Use of Commonly Owned Standards

8) Licensing

9) Permanence

10) Usage Costs

2.2 Cassandra

Bei Cassandra handelt es sich um eine spaltenorientierte NoSQL-Datenbank. Der Begriff des spaltenorientierten Datenbankmanagementsystems (DBMS) bezieht sich zunächst auf die Art und Weise, in der die Daten auf dem verwendeten Speichermedium abgelegt werden. Traditionell zeilenorientierte Datenbanken legen alle Felder (Attribute) einer Tabellenzeile (Tupel) hintereinander ab. Dies ist von Vorteil, wenn in einer Transaktion wenige komplette Tupel gelesen oder geschrieben werden sollen.

Im Gegensatz dazu kann bei einer rein spaltenorientierten Datenbank jede Spalte in einer eigenen Datei liegen, das heißt auf einen Wert eines Attributes eines Tupels folgt in Lese-Reihenfolge nicht das nächste Attribut des selben Tupels, sondern das gleiche Attribute des nächsten Tupels. In der folgenden Abbildung wird diese Form der Speicherung skizziert:

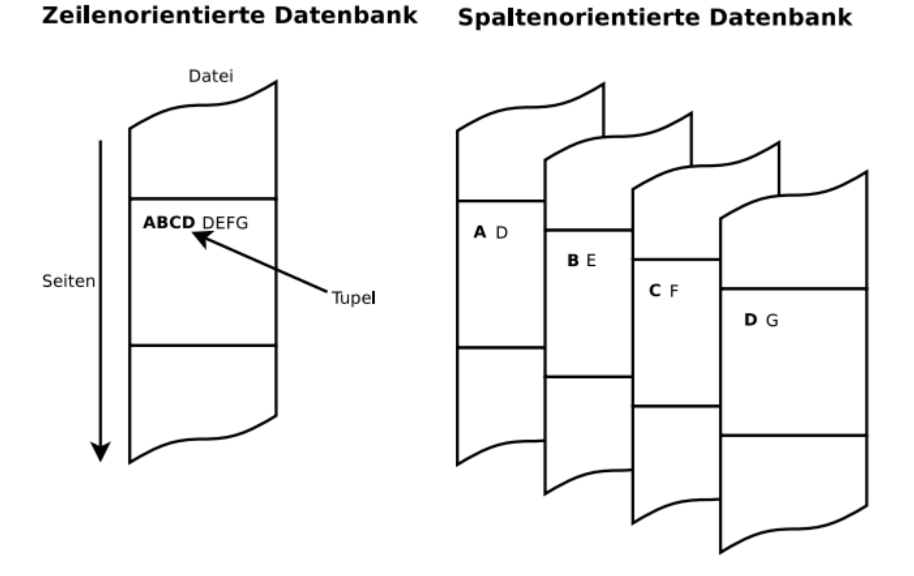


Abb. 2: Speicherung der zwei Tupel {A,B,C,D} und {D,E,F,G} in verschiedenen physischen Datenmodellen[9]

[9] Mit Änderungen entnommen: Bösswetter, D. (2010)

Ihren Einsatz finden spaltenorientierte DBMS hauptsächlich beim Online Analytical Processing (OLAP), also der Auswertung von bereits erhobenen Daten. Intuitiv funktioniert dies, da hier meistens wenige Attribute von sehr vielen Tupeln benötigt werden und durch die Spaltenorientierung die restlichen Attribute nicht gelesen werden müssen. Eine reine Spaltenorientierung reicht jedoch nicht aus, um große Fortschritte bei der Anfrage-Bearbeitung zu erreichen. So bleibt die Benutzerschnittstelle im Vergleich zu relationalen Datenbanken zwar weiterhin SQL, die dahinter liegende Ausführungsplanung und -optimierung ändert sich aber. Bei analytischen Anfragen werden oft, um Werte zu verdichten, viele Datensätze benötigt. Da das sequentielle Lesen der Daten aus technischen Gründen meistens schnell geht und eventuelle Kompression wahlfreie Zugriffe erschwert, hat es sich als effektive Strategie etabliert, mehrere Spalten parallel zu scannen und somit immer die benötigten Attribute in einem schnellen, flüchtigen Speicher zu haben. Operatoren werden nicht mehr als Iteratoren über einzelne Tupel realisiert, sondern als Iteratoren über ganze Blöcke von Werten.

Weiterhin ändern sich einige Optimierungsziele, wenn ein spaltenorientiertes Datenmodell zugrunde gelegt wird. Während bei relationalen Datenbankmanagementsystemen (RDBMS) gilt, dass die Projektionsoperatoren, die die benötigten Attribute aus allen verfügbaren herausprojizieren, möglichst früh auszuführen sind, wird in einer spaltenorientierten Datenbank daraus eine späte Materialisierung, da dann die benötigten Attribute erst so spät wie möglich hinzu geladen werden sollten.

Nicht nur die Spaltenorientierung sollte so lange wie möglich während der Ausführung beibehalten werden, sondern auch die Kompression. Untersuchungen haben gezeigt, dass ein Column-Store, der komprimierte Spalten liest und sofort zu unkomprimierten Relationen zusammensetzt, sich ähnlich wie eine zeilenorientierte Datenbank verhält.[10]

2.3 DynamoDB

Bei DynamoDB handelt es sich um eine NoSQL Datenbank die Eigenschaften von *key value stores* und *document stores* vereint. Diese wurde von Amazon entwickelt und ist ausschließlich über die Cloud Plattform Amazon Web Services (siehe Kapitel 4) verfügbar. Zu Testzwecken kann DynamoDB ebenso lokal auf einem Computer betrieben werden – dazu dient das Projekt DynamoDB Local. DynamoDB wird komplett von Amazon verwaltet, d. h. die Einrichtung, die Beschaffung der Hardware sowie beispielsweise Wartung der Datenbank müssen nicht vom Kunden durchgeführt werden.[11]

[10] Vgl. Bösswetter, D. (2010)
[11] Vgl. Vogels, W. (2012)

Key value stores dienen zur Speicherung von schemalosen Daten. Grundsätzlich können beliebige Werte in eine solche Datenbank gespeichert werden. Diese Werte werden jeweils über einen Key angesprochen.[12]

Document stores erweitern die Idee der *key value stores* und dienen zur Speicherung von Werten in einem definierten Format, beispielsweise JavaScript Object Notation (JSON). Dadurch ist es möglich gespeicherte Elemente nicht nur anhand des Keys sondern auch ebenso anhand der Attribute der gespeicherten Dokumente abzufragen.

DynamoDB kennt zwei verschiedene Key-Typen: Hash-Schlüssel und zusammengesetzte Hash-Bereichsschlüssel. Ein Hash-Schlüssel besteht dabei aus einem einzigen Attribut, beispielsweise *UserID*. Ein zusammengesetzter Hash-Bereichsschlüssel beschreibt einen Index über ein Hash-Schlüsselelement und ein Bereichsschlüsselelement, beispielweise *UserID* und *Zeitstempel.* In einer Tabelle mit Bestellungen kann so performant nach allen Bestellungen des Users mit der *UserID* 3 innerhalb der letzten 24h gesucht werden.[13] Ein Eintrag, ein sogenanntes Item, in einer DynamoDB-Tabelle besteht folglich aus einem Hash-Schlüssel, einem optionalen Bereichsschlüssel und verschiedenen Attributen. Dies ermöglicht eine schnelle Abfrage von Items anhand dieser beiden Schlüssel. Sollen Items, die im JSON-Format gespeichert sind, auch über ihre Attribute abgefragt werden können, können *Global Secondary Indexes* oder *Local Secondary Indexes* definiert werden.[14]

[12] Vgl. Seeger, M. (2009), S. 5
[13] Vgl. Amazon Web Services (2015d)
[14] Vgl. Amazon Web Services (2014a)

3 Open Data

In diesem Kapitel und den dazugehörigen Unterkapiteln wird die Beschaffung geeigneter Testdatensätze für die beiden Datenbankmodelle beschrieben. Geplant ist dabei, die benötigten Datensätze aus offenen Beständen (sog. Open Data) abzuziehen. Aus diesem Grund werden zunächst beispielhafte Bestände vorgestellt, die im Internet gefunden werden können. Hier wird auch auf deren Eignung für das Projekt eingegangen. Im Anschluss daran erfolgt die Auswahl geeigneter Testdaten.

3.1 Datenbestände im Internet

Eine Stichwortsuche nach Open Data im Internet bringt eine Vielzahl unterschiedlichster Anbieter und Datentypen hervor. Grundsätzlich zeigt sich, dass offene Datenbestände vor allem von Regierungen, Ländern oder Kommunen herausgegeben werden. Hierbei handelt es sich, wie bereits beschrieben (siehe Kapitel 1.1), um Open Government Data. Da im Rahmen dieser Arbeit nicht die komplette Vielfalt vorgestellt werden kann, erfolgt im Folgenden die Präsentation von vier Anbietern. Diese werden auch hinsichtlich ihrer Eignung für das Projekt untersucht.

3.1.1 UK Government

Die britische Regierung ist einer der größten Anbieter offener Daten. Die Veröffentlichung dieser Daten erfolgt über die Domain *data.gov.uk*. Diese befindet sich momentan noch in einer Beta-Version. Ziel ist aber, die gesammelten Datenbestände aller Ministerien, Abteilungen und Kommunen über diese eine Plattform verfügbar zu machen. Bereits jetzt werden über *data.gov.uk* rund 20.000 unterschiedlicher Datenbanken angeboten.[15] Die Suche nach geeigneten Datensätzen erweist sich hier generell als einfach. Neben der Möglichkeit der Filterung des Angebots nach Thema (z.B Society, Environment, Health,…), kann auch das favorisierte Datenformat ausgewählt werden. Die meisten Datenbestände werden in mehreren Formaten angeboten (z.B. CSV, XLS, PDF, HTML,…).

Der klar definierte Zweck dieser Veröffentlichungen ist, den Bürgerinnen und Bürgern des Landes Zugang zu wichtigen Indikatoren zu verschaffen. Dadurch soll das Verständnis für politische Entscheidungen gefördert werden.[16] Die Bereitstellung von Informationen für den Normalverbraucher steht hier im Vordergrund. Die Suche nach Datenbeständen für eine wie in diesem Projekt benötigte Testdatenbank gestaltet sich deutlich schwerer.

Dies liegt vor allem an der fehlenden Möglichkeit einer Filterung nach Größe beziehungsweise Anzahl der Datensätze. So werden für Performance-Tests der

[15] Vgl. data.gov.uk (o.J.a)
[16] Vgl. data.gov.uk (o.J.b)

Datenbanken mehrere Millionen Datensätze benötigt. Es lässt sich bei der britischen Regierung jedoch kein Datenbestand finden, der dieser Größe entspricht.

Inzwischen bietet die deutsche Bundesregierung ein Äquivalent zum britischen *data.gov.uk.* Über die Domain *govdata.de* sollen dabei die Daten aller Verwaltungsebenen zentral zugänglich gemacht werden.[17] Im Moment wird auch dieses Portal nur testweise betrieben. Mit einem Angebot von etwa 9.700 unterschiedlichen Datenbanken, bietet es jedoch wesentlich weniger offene Daten an, als das britische Pendant.[18]

3.1.2 Senatsverwaltung Berlin

Wie bereits zuvor beschrieben (siehe Kapitel 1.2.1), ist die Sammlung aller offenen Datenbestände der deutschen Verwaltungseinrichtungen noch nicht abgeschlossen. Nichtsdestotrotz bieten viele Abteilungen und Kommunen im Bundesgebiet Daten zur freien Nutzung an. Anbieter sind hier besonders die großen Kommunen, wie Köln oder Berlin.[19] Kennzahlen und Indikatoren der Stadt Berlin werden von der *Senatsverwaltung für Wirtschaft, Technologie und Forschung* herausgegeben. Das Portal unter der Domain *daten.berlin.de* ist ähnlich der Seite der britischen Regierung aufgebaut (siehe Kapitel 1.2.1). Die Anzahl unterschiedlicher Datenquellen ist aber um ein vielfaches geringer. Es besteht eine ungefähre Auswahl zwischen 250 und 300 Datenbanken. Die exakte Größe des Angebots ist nicht beziffert. Im Gegensatz zu *data.gov.uk* besteht auch nicht die Möglichkeit der Filterung nach Datenformat. Die meisten Datenbanken sind jedoch in mehreren Formaten abrufbar.[20] Auch das Angebot der Senatsverwaltung in Berlin bietet keine ausreichende Datenmenge für Performancetests von DynamoDB und Cassandra. Die vorhandenen Daten dienen meist nur der Informationsgewinnung für Bürgerinnen und Bürger. Eine beispielhafte Datenquelle ist die Auflistung aller Berliner Weihnachtsmärkte im Jahr 2014 (siehe *Tab. 1*). Insgesamt enthält diese Datenbank 103 Datensätze.

[17] Vgl. Finanzbehörde Hamburg (o.J.a)
[18] Vgl. Finanzbehörde Hamburg (o.J.b)
[19] Vgl. Stadt Köln (o.J.)
[20] Vgl. Senatsverwaltung für Wirtschaft, Technologie und Forschung (o.J.)

Marktname	Straße	PLZ + Ort	Beginn
Weihnachtsmarkt vor dem Schloss Charlottenburg	Spandauer Damm zwischen Luisenplatz und Klausenerplatz	14059 Berlin	24.11.2014
31. Weihnachtsmarkt an der Gedächtniskirche	Breitscheidplatz	10789 Berlin	24.11.2014
Weihnachtsmarkt in der Fußgängerzone Wilmersdorfer Straße	Wilmersdorfer Straße	10627 Berlin	28.11.2014
Weihnachten in Westend (teilgewerblich)	Preußenallee	14052 Berlin	20.12.2014
Weihnachtsmarkt Berlin-Grunewald des Johannischen Sozialwerks e. V.	St.-Michaels-Heim Bismarckallee 23	14193 Berlin	28.11.2014

Tab. 1: Beispieldatensätze der Senatsverwaltung Berlin[21]

3.1.3 Bibliotheksverbund Bayern

Die zuvor betrachteten Anbieter offener Daten konnten die für das Projekt benötigte Menge an Datensätzen nicht aufbringen (siehe Abschnitte 1.2.1 und 1.2.2). Eine Recherche hat hier ergeben, dass nur wenige Anbieter Datenmengen in der Größe mehrerer Millionen Sätze zur Verfügung stellen. Eine Ausnahme stellt der *Bibliotheksverbund Bayern* dar. In Kooperation mit dem *Bibliotheksverbund Berlin-Brandenburg*, werden rund 23 Millionen bibliografische Datensätze unterschiedlicher Universitäten, Hochschulen oder Bibliotheken zum Download angeboten. Das Gesamtangebot wird hierbei in separat herunterladbaren ZIP-Dateien zur Verfügung gestellt. Der *Bibliotheksverbund Bayern* stellt die Daten dabei lediglich im XML-Format zur Verfügung.[22]

Trotz ausreichender Menge an Datensätzen, ist auch das Angebot des Bibliotheksverbundes untauglich für den Testbetrieb der beiden Datenbankmodelle. Zwar sind Performance-Tests möglich, die vorhandenen Daten sind jedoch bibliotheksspezifisch und nicht vergleichbar mit dem Datenaufkommen der Versicherungsbranche. Da dieses Projekt auf den Einsatz von NoSQL-Lösungen in eben dieser Branche abzielt, muss auch der Testbetrieb möglichst realitätsnah gestaltet sein. Aus diesem Grund wird davon abgesehen, die eben beschriebenen bibliografischen Datensätze im weiteren Testbetrieb anzuwenden.

3.1.4 World Bank Group

Neben rein nationalen Datenquellen, gibt es auch internationale Zusammenschlüsse, die Open Data anbieten. Ein solcher Zusammenschluss ist die *World Bank Group* mit Sitz in Washington D.C. Die World Bank ist keine Bank im eigentlichen Sinne, sondern eine Organisation zur Minderung von Armut in Entwicklungsländern und zur Förderung der wirtschaftlichen Entwicklung.

[21] Enthalten in: Senatsverwaltung für Wirtschaft, Technologie und Forschung (2014)
[22] Vgl. Bibliotheksverbund Bayern (2014)

Im Zuge dessen sammelt die Institution eine Vielzahl an Daten über einzelne Staaten. Diese stehen auf der Domain *worldbank.org* zur freien Verfügung. Insgesamt werden mehrere hundert unterschiedliche Datenbanken angeboten, die wiederum mittels Filterfunktion ausgewählt werden können. Da die meisten der angebotenen Datenquellen in einer bestimmten Periode aktualisiert werden, stehen meist mehrere Versionen der Daten zur Verfügung.[23] Zu den größten verfügbaren Datensammlungen gehören die World Development Indicators. Das ist eine Sammlung länderspezifischer Indikatoren, die jährlich von der *World Bank* veröffentlicht wird. Diese Datenbank umfasst über 300.000 Datensätze.[24]

Im Vergleich zu den zuerst präsentierten Anbietern offener Daten (siehe Abschnitte 1.2.1 und 1.2.2), bietet die *World Bank* demnach eine größere Zahl an Datensätzen. Um die beiden Datenbanksysteme an ihre Grenzen führen zu können, ist jedoch auch diese Zahl zu gering.

3.2 Wahl von Testdaten für die NoSQL-Datenbank

Die Suche nach geeigneten Testdaten durch Recherche im Internet lieferte kein zufriedenstellendes Ergebnis. Erstens erwies es sich als schwer eine große Datenmenge aufzutreiben und zweitens findet man nur wenig offene Datensätze, die mit versicherungsspezifischen Daten vergleichbar sind. Innerhalb eines vertretbaren Zeitrahmens eine Datenquelle zu finden, die beide Voraussetzungen erfüllt, ist Stand heute nicht möglich.

Um trotz fehlender freier Datenbestände einen Performance- und Funktionstest der beiden Datenbankmodelle durchführen zu können, muss ein eigener Testdatenbestand aufgebaut werden. Die Schritte dazu werden in den kommenden Abschnitten aufgeführt. Ziel ist ein Bestand mit mehreren Millionen personenbezogenen Datensätzen. Die Wahl von personenbezogenen Daten als Testobjekte erfolgte aufgrund der Vergleichbarkeit zu Versicherungsbeständen und der Verfügbarkeit von Namens-, Straßen- und Städtelisten im World Wide Web. Anbieter solcher Listen findet man zahlreiche. Listen dieser Art werden meist vom privaten Sektor zum Download angeboten. Für die Straßennamen der Testdatensätze wurde beispielsweise die Liste der *wer denkt was GmbH* verwendet.[25]

Mithilfe einer Java-Applikation können die einzelnen Listen miteinander verknüpft werden. Dazu müssen die Listen der Einfachheit halber zunächst in das Comma-separated values-Format (CSV) konvertiert werden. Der genaue Aufbau des Programms kann Anhang 1

[23] Vgl. World Bank Group (o.J.)
[24] Vgl. World Bank Group (2014)
[25] Vgl. Wer denkt was GmbH (2013)

entnommen werden. Durch die Verknüpfung der CSV-Dateien werden für dieses Projekt über 20 Millionen unterschiedliche Datensätze produziert.

Bei Mehrbedarf könnten durch kleine Anpassungen im Code beliebig große Datenmassen erzeugt werden. Ein einzelner Datensatz soll die folgenden Elemente enthalten:

- Personennummer
- Nachname
- Vorname
- Geburtsdatum
- Stadt
- Straße
- Hausnummer
- Vertragsnummer

4 Amazon Web Services

Amazon Web Services (AWS) ist die Cloud Computing Plattform von Amazon.com. Diese besteht seit dem Jahr 2006 und hat sich seitdem zu einer „hoch verfügbare[n], skalierbare[n] und kostengünstige[n] Rechenplattform in der Cloud“[26] entwickelt. Im Folgenden wird ein Überblick über die Plattform gegeben und näher auf die Verwendung eines der angebotenen Produkte, den Datenbankservice DynamoDB, eingegangen.

4.1 Verfügbarkeit

Die Services von AWS werden in elf verschiedenen geographischen Regionen angeboten: USA Ost (Nord-Virginia), USA West (Oregon und Nordkalifornien), AWS GovCloud (USA), Brasilien (São Paulo), Europa (Irland und Frankfurt), Asien-Pazifik (Singapur, Sydney und Tokio) sowie China (Peking). Jede Region ist in sich geschlossen, d. h. Daten verlassen nie die jeweilige Region. In jeder dieser Regionen existieren mehrere sogenannte *Availablity Zones,* die dazu verwendet werden, die Verfügbarkeit der angebotenen Leistungen durch Replikation zu erhöhen.[27]

4.2 Angebot

AWS bietet eine Vielzahl von verschiedenen Services in diversen Anwendungsgebieten an:

- Datenverarbeitung
- Speicherung und Bereitstellung von Inhalten
- Datenbanken
- Networking
- Administration und Sicherheit
- Analysen
- Anwendungsservices wie Push-Benachrichtigungen oder E-Mail-Zustellung
- Deployment

Im Rahmen dieser Ausarbeitung wird genauer auf die Verwendung des NoSQL-Datenbankservices DynamoDB eingegangen.[28]

[26] Vgl. Amazon Web Services (2015a)
[27] Vgl. Amazon Web Services (2015b)
[28] Vgl. Amazon Web Services (2015c)

4.3 DynamoDB

4.3.1 Preismodell

Kosten, die bei der Verwendung von DynamoDB anfallen, richten sich nach der bestellten Durchsatzkapazität, nicht nach der tatsächlichen Nutzung. D. h. es muss im Vorfeld durch den Kunden berechnet werden, welcher Durchsatz erwartet wird. Dies erfolgt durch die Ermittlung von zwei Größen: Schreibkapazitätseinheiten und Lesekapazitätseinheiten.[29]

Mit einer Schreibkapazitätseinheit kann ein Schreibvorgang pro Sekunde für Elemente mit einer Größe von bis zu einem KB durchgeführt werden. Eine Lesekapazitätseinheit dient dazu einen Lesevorgang pro Sekunde für Elemente mit einer Größe von bis zu vier KB auszuführen. Sind die zu lesenden bzw. schreibenden Elemente kleiner als ein KB, so kann mit jeder Kapazitätseinheit nur ein Lese- bzw. Schreibvorgang pro Sekunde durchgeführt werden. Sind die Elemente größer als ein KB kann die benötigte Kapazität mit folgenden zwei Formeln bestimmt werden:

„Für Schreibvorgänge erforderliche Kapazitätseinheiten = Anzahl der Elementschreib-vorgänge pro Sekunde x Elementgröße in KB (aufgerundet auf die nächste ganze Zahl)

Für Lesevorgänge erforderliche Kapazitätseinheiten = Anzahl der Elementlesevorgänge pro Sekunde x Elementgröße in KB (aufgerundet auf die nächste ganze Zahl)“

Zu beachten ist, dass der Verbrauch von Kapazitätseinheiten an der Anzahl der gelesenen/geschriebenen Elemente gemessen wird und nicht an der Anzahl der API-Aufrufe, die dazu nötig waren.

Für die Speicherung der Daten selbst fallen ebenfalls Kosten an: die ersten 25 GB pro Monat sind kostenlos, darüber hinaus kostet jedes zusätzliche GB pro Monat extra.

In folgenden sind die Preise für die Region Frankfurt angegeben[30]:

- Schreibdurchsatz: $0.00793 pro Stunde für je 10 Schreibkapazitätseinheiten
- Lesedurchsatz: $0.00793 pro Stunde für je 50 Lesekapazitätseinheiten
- Speicherplatz: $0.306 pro GB und Monat, falls die ersten 25 GB überschritten wurden

4.3.2 Einrichtung

Eine DynamoDB Tabelle kann über die Amazon Management Console angelegt werden. Dazu führt ein Einrichtungsassistent (siehe Abb. 1) durch die notwendigen Schritte. Alle folgenden Operationen wurden in der Region Frankfurt durchgeführt.

[29] Vgl. Amazon Web Services (2015d)
[30] Vgl. Amazon Web Services (2015e)

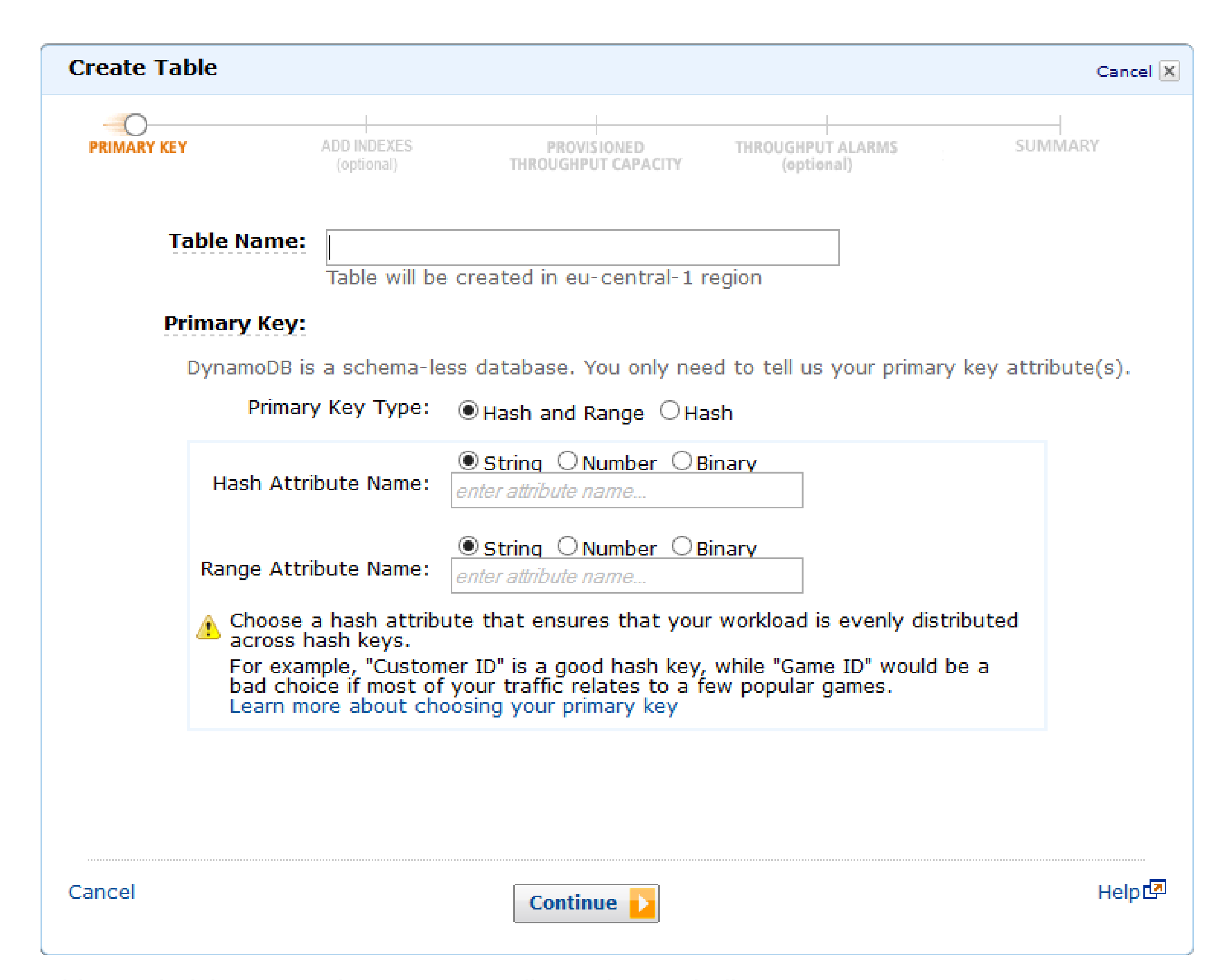


Abb. 3: Einrichtungsassistent zur Erstellung einer Tabelle

Im ersten Schritt muss der Tabellenname und der Primärschlüssel der Tabelle festgelegt werden. Dabei kann der Primärschlüssel in zwei verschiedenen Typen vorliegen: Hash-Schlüssel oder zusammengesetzter Hash-Bereichsschlüssel.

Danach können Indexe für Attribute von Items in der Tabelle festgelegt werden. Indexe können nur während der Erstellung einer Tabelle angelegt werden. Sobald die Tabelle erzeugt wurde ist es nicht mehr möglich Indexe zu definieren, zu ändern oder zu entfernen.

Im dritten Schritt müssen die Lese- und Schreibkapazitätseinheiten der Tabelle festgelegt werden. Diese Werte können nach der Tabellenerstellung beliebig erhöht, jedoch nur zweimal pro Tag verringert werden.

Im vorletzten Schritt kann eine Überwachungsfunktion aktiviert werden. Ist diese aktiviert so werden Statusnachrichten an eine beliebig definierbare E-Mail-Adresse versendet, sobald die Kapazitätseinheiten zu einem bestimmten Prozentsatz in Anspruch genommen werden. Für diesen Überwachungsservice, der über Amazon CloudWatch bereitgestellt wird, können zusätzliche Kosten anfallen (siehe Anhang 1).

Der letzte Schritt gibt eine Übersicht darüber, wie die Tabelle angelegt werden soll. Sind die Daten korrekt kann die Tabelle mit einem Klick auf „Create“ erzeugt werden. Die Erzeugung dauert wenige Minuten und variiert je nach bereitzustellender Durchsatzkapazität.

Die Tabelle, die in Kapitel 3.2 beschrieben ist, wurde mit folgenden Werten angelegt:

Tabellenname	Kunden
Primärschlüssel	Hash-Schlüssel über String-Attribut *persnr*
Indexe	-
Durchsatzkapazität	Je 25 Lese- und Schreibkapazitätseinheiten
Überwachungsfunktion	Nicht aktiviert

Tab. 2: Erzeugungsinformationen der Tabelle in DynamoDB

4.3.3 Befüllung

Daten können über eine „Import/Export“-Funktion in die Tabelle eingespielt werden – diese Funktion ist aus, den Autoren dieses Dokuments, nicht bekannten Gründen nicht für die Region Frankfurt verfügbar. Aus diesem Grund werden die Daten mithilfe eines selbst entwickelten Java-Programmes (siehe Anhang 2) in die Tabelle *Kunden* geschrieben. Dazu werden die Daten, die im CSV-Format vorliegen mithilfe der Open-Source Bibliothek *opencsv* eingelesen und mit dem AWS Software Development Kit (SDK) für Java in die Datenbank eingespielt. Es werden jeweils 25 Elemente per *BatchWriteItemRequest* geschrieben – das Maximum an Elementen pro Request. Insgesamt hat das Einspielen der Daten ~21h in Anspruch genommen. Dies entspricht ~300 Schreibvorgängen pro Sekunde.

4.3.4 Verwendung

Einfache Lese- und Schreiboperationen können über die Amazon Management Console durchgeführt werden. Es können neue Elemente mithilfe eines simplen Editors (siehe Abb. 2) in die Tabelle eingefügt werden. Außerdem lassen sich Elemente anhand ihres Primärindex und ihrer Sekundärindexe suchen. Für komplexere Abfragen und Anforderungen ist die Verwendung der DynamoDB API erforderlich. Eine Verwaltung der Tabelle mit einer SQL-ähnlichen Sprache ist von Amazon nicht vorgesehen, kann jedoch durch die Verwendung von Produkten von Drittanbietern, wie beispielsweise RazorSQL, erreicht werden.

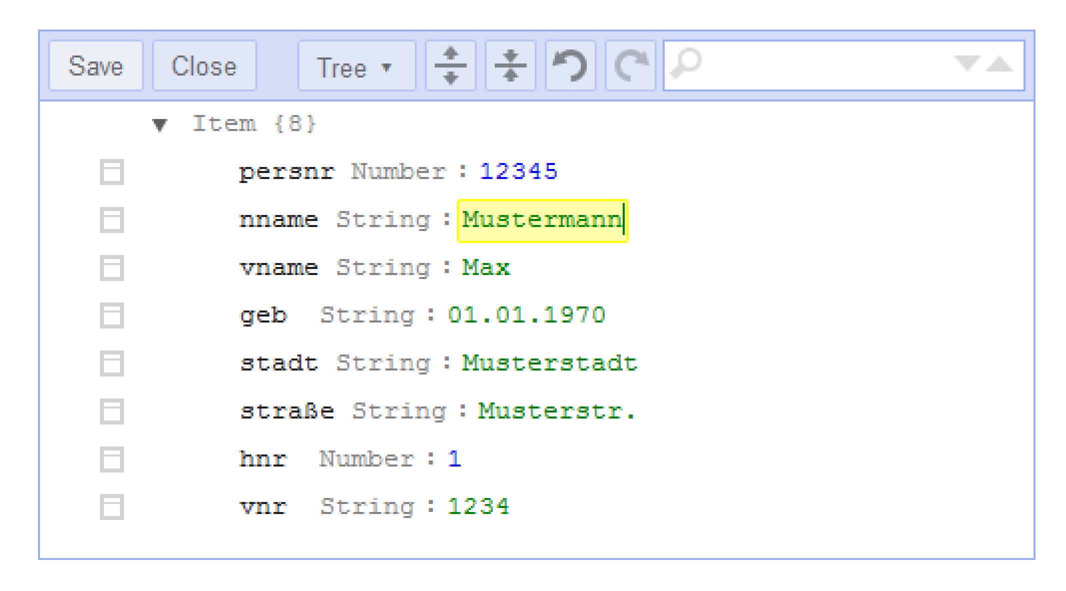


Abb. 4: Item-Editor für neue Elemente einer DynamoDB-Tabelle

4.3.5 Zugriffskontrolle

Um außerhalb der Amazon Management Console mit Tabellen der DynamoDB arbeiten zu können, muss zunächst ein User über *Identity and Access Management* (IAM) angelegt werden. Dieser erhält eine *Access Key ID* und einen *Secret Access Key* zur Authentifizierung. User können mithilfe von *User* und *Group Policies* Zugriff auf bestimmte Bereiche des jeweiligen AWS Kontos erhalten. So ist es beispielsweise möglich einem User Vollzugriff auf die Tabelle *Kunden* zu gewähren und einem anderen User dagegen nur lesenden Zugriff. Usern kann der Zugriff jederzeit wieder entzogen werden.[31,32]

```
{
    "Version": "2014-12-13",
    "Statement": [
        {
            "Action": [
                "dynamodb:*"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Abb. 5: Berechtigungen des Users des Datenimportprogramms

Für das Datenimportprogramm aus Abschnitt 4.3.3 wurde ein User mit den in Abb. 5 dargestellten Berechtigungen angelegt. Diese sorgen dafür, dass der User Vollzugriff auf alle DynamoDB-Tabellen erhält.

[31] Vgl. Amazon Web Services (2015f)
[32] Vgl. Amazon Web Services (2015g)

4.3.6 DynamoDB API

Das DynamoDB Application Programming Interface (API) kann über das AWS SDK verwendet werden. Dieses wird für drei Programmiersprachen angeboten: Java, .NET und PHP. Diese abstrahieren den Zugriff auf das DynamoDB API und vereinfachen so den Zugriff auf die Tabellen. Das DynamoDB API ist eine REST-Schnittstelle und arbeitet mit einer speziellen JSON-Datenstruktur um Anfragen und Antworten zu beschreiben.

Das DynamoDB API unterstützt folgende Operationen:

Operation	Beschreibung
BatchGetItem	Mehrere Items lesen
BatchWriteItem	Schreibt mehrere Items
CreateTable	Erstellen einer neuen Tabelle
DeleteItem	Löschen eines Items anhand dessen Primärschlüssels
DeleteTable	Entfernen einer Tabelle und deren Inhalts
DescribeTable	Liefert Strukturinformationen zur Tabelle
GetItem	Liest ein Item anhand dessen Primärschlüssels
ListTables	Listet mit dem AWS Account verbundene Tabellen auf
PutItem	Schreibt ein neues Item oder ersetzt ein bestehendes
Query	Abfrage von Items über deren Indexe
Scan	Abfrage über alle Items der Tabelle
UpdateItem	Ändert Attribute eines bestehenden Items oder legt ein neues an
UpdateTable	Änderung der Lese- und Schreibkapazitätseinheiten

Tab. 3: Operationen des DynamoDB API[33]

[33] Vgl. Amazon Web Services (2014b): S. 9

5 Aufbau einer Cassandra-Datenbank aus der Cloud

In diesem Kapitel wird der Aufbau einer Cassandra-Datenbank aus der Cloud beschrieben. Als Cloud-Provider wurde im ersten Versuch Microsoft Azure und im zweiten Versuch Google Cloud verwendet. Die nachfolgenden Kapitel beschreiben und dokumentieren den Verlauf der beiden Versuche.

5.1 Erster Versuch: Cassandra auf Microsoft Azure

Um den Aufbau von Cassandra auf dem Cloud-Provider Microsoft Azure besser nachvollziehen zu können, wird zuerst der Provider sowie dessen Plattform vorgestellt. Anschließend werden die verwendeten Hilfsprogramme definiert, sodass im dritten Teil dieses Kapitels der Aufbau von Cassandra auf Microsoft Azure beschrieben werden kann. Das anschließende Fazit des ersten Versuches schließt dieses Kapitel ab.

5.1.1 Der Cloud-Provider Microsoft Azure[34]

Microsoft Azure ist, wie der Name bereits sagt, die Cloud-Plattform von Microsoft. Sie bietet eine Sammlung integrierter Dienste, zum Beispiel frei nutzbarer Speicher, Networking und das Verwalten von Apps. Zudem ist Azure die einzige große Cloud-Plattform, die von Gartner (einem Anbieter, der Marktforschungsergebnisse und Marktanalysen über die Entwicklungen in der IT anbietet) als Marktführer sowohl für Infrastructure-as-a-Service (IaaS) als auch für Platform-as-a-Service (PaaS) geführt wird. Mit dieser Kombination aus verwalteten und nicht verwalteten Diensten können Anwendungen nach Wunsch erstellt, bereitgestellt und verwaltet werden.

Microsoft Azure unterstützt alle Betriebssysteme, Sprachen, Tools und Frameworks – von Windows bis Linux, SQL Server bis Oracle, C# bis Java. So können Anwendungen und Dienste erstellt werden, die mit jedem Gerät funktionieren.

Auf Seiten der Erreichbarkeit und des Supports bietet Azure ein Service-Level-Agreement (SLA) von 99,95% sowie technischen Support und Integritätsüberwachung rund um die Uhr. Mit einem globalen Netzwerk aus Microsoft-Rechenzentren in 13 Regionen bietet Microsoft zudem Möglichkeiten die Leistung von Anwendungen zu optimieren.

5.1.2 Verwendete Hilfsprogramme

Für den Aufbau von Cassandra auf Microsoft Azure mussten einige Hilfsprogramme hinzugezogen werden:

- **PuTTY**

PuTTY ist eine Open-Source Software zum Herstellen von Verbindungen über Secure Shell,

[34] Vgl. Microsoft Corporation (o.J.b)

Telnet, Remote login oder serielle Schnittstelle. Dabei dient PuTTY als Client und stellt die Verbindung zu einem Server her. Beim Verbindungsaufbau wird die Identität des Benutzers mittels einer der bereitgestellten Methoden zur Authentifizierung überprüft. PuTTY ist für Windows und Linux verfügbar.[35] Die Einstiegsmaske kann in Anhang 4 eingesehen werden. In Abb. 6 wird die Konsole, die nach erfolgreichem Verbindungsaufbau erscheint, dargestellt.

```
dennis@raspberrypi: ~
login as: dennis
dennis@192.168.2.43's password:
Linux raspberrypi 3.12.22+ #691 PREEMPT Wed Jun 18 18:29:58 BST 2014 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Dec 29 11:35:29 2014 from 192.168.2.38
dennis@raspberrypi ~ $
```

Abb. 6: PuTTY Konsole

- **WinSCP**

Windows Secure Copy (WinSCP) ist ein grafischer SFTP-Client für Windows auf der Basis von SSH und ermöglicht einen geschützten Daten- und Dateitransfer zwischen verschiedenen Rechnern. Dem Benutzer steht dabei eine Oberfläche zur Verfügung, die dem Windows-Explorer ähnelt. WinSCP steht unter der GNU General Public License und ist damit freie Software. Der SSH- und SCP-Programmcode basiert auf PuTTY, der FTP-Programmcode basiert auf FileZilla.[36] Die folgende Abbildung zeigt eine beispielhafte Verbindung mit einem Server.

[35] Vgl. Wikipedia (2014a)
[36] Vgl. Wikipedia (2014b)

Abb. 7: WinSCP – aktive Sitzung

❖ **Node.js**

Node.js ist eine serverseitige Plattform zum Betrieb von Netzwerkanwendungen und basiert auf der JavaScript-Laufzeitumgebung „V8", die ursprünglich für Google Chrome entwickelt wurde. Sie bietet eine ressourcensparende Architektur, die eine besonders große Anzahl gleichzeitig bestehender Netzwerkverbindungen ermöglicht.[37]

Für dieses Projekt verwenden wir Node.js hauptsächlich zur serverseitigen Ausführung von JavaScript-Dateien.

5.1.3 Aufbau von Cassandra auf Microsoft Azure

Der Aufbau von Cassandra auf Microsoft Azure wurde anhand einer Anleitung von Microsoft durchgeführt. Im Folgenden werden daher nicht auf die einzelnen Befehle eingegangen, sondern nur der Vorgang an sich beschrieben. Zur besseren Verständlichkeit der folgenden Schritte, ist in Anhang 4 die Einstiegsmaske von Azure beigefügt.

Im ersten Schritt muss auf der Azure-Plattform eine „virtual machine" (VM) erstellt werden. Azure bietet hierbei die Möglichkeit aus verschiedenen Betriebssystemen das passende System auszuwählen. Für die NoSQL-Datenbank Cassandra wählen wir das Betriebssystem „Ubuntu Server" aus. Nach Vergabe einer Bezeichnung der VM stellt Microsoft Azure nach kurzer Aufbauzeit die virtuelle Maschine zur Verfügung.

[37] Vgl. Wikipedia (2014c)

Im zweiten Schritt muss nun die VM konfiguriert werden. Erforderlich sind hierbei ein neuer Benutzername sowie ein neues Kennwort. Das Sichern der Authentifizierung mit einem SSH-Schlüssel ist ebenfalls möglich.

Da die Dokumentation ein Cassandra-Cluster empfiehlt, müssen die ersten beiden Schritte für weitere virtuelle Maschinen erneut durchgeführt werden.

Nun soll der Anleitung entsprechend eine der VM's die Funktion als Thrift-Endpunkt mit Lastenausgleich zugeordnet werden. Dies hat leider aufgrund der spärlichen und teilweise veralteten Dokumentation nicht funktioniert. Da auch nach einigen Stunden an Testen keine Lösung hierfür gefunden werden konnte, fiel die Entscheidung darauf, die Cassandra-Datenbank auf nur einer virtuellen Maschine aufzubauen.

Die Installation von Cassandra auf einer virtuellen Maschine von Azure muss manuell durchgeführt werden. Bis auf ein paar kleinere Fehler in der dokumentierten Befehlsreihenfolge der Anleitung war dies unkompliziert, jedoch zeitaufwendig.

Um später auf die Datenbank zugreifen zu können, wird von Microsoft eine weitere VM empfohlen, die als Client-Server fungieren soll. Der Client-Server wird ebenfalls mit dem Betriebssystem „Ubuntu Server" aufgesetzt. Um JavaScript Dateien performant ausführen zu können, wird zudem die bereits angesprochene Software Node.js installiert. Um später eine Verbindung zur Cassandra-Datenbank herstellen zu können, müssen zusätzlich einige Pakete heruntergeladen werden. Die Anleitung funktionierte hierbei ohne größere Probleme.

Um Testdatensätze auf die NoSQL-Datenbank zu bekommen, wird ein bereits dokumentiertes Node.js-Skript vorgeschlagen. Leider wies dies einige Fehler auf, sodass Recherchearbeiten nötig waren. Nachdem der JavaScript-Interpreter keine Fehlermeldungen mehr meldete, zeigte sich jedoch ein weiteres Problem. Die Befehle die vom Client-Server an die virtuelle Maschine der Cassandra-Datenbank gesendet wurden, erreichten die Datenbank nicht. Der Grund hierfür lag in der Portverwaltung der virtuellen Maschinen. Aus unerklärbaren Gründen konnte der Eingangsport der Cassandra-Datenbank nicht erreicht werden.

Da hiermit keine Chance mehr bestand, die NoSQL-Datenbank in der vorgegeben Zeit einzurichten, wurde der Versuch, Cassandra auf Microsoft Azure aufzubauen, abgebrochen.

5.1.4 Fazit

Microsoft Azure bietet an sich viele Möglichkeiten seine Projekte anforderungsgerecht umzusetzen. Das Verwaltungsportal ist sehr übersichtlich gestaltet. Die Möglichkeit aus vielen verschiedenen Betriebssystemen das für sich passende System herauszufinden, ist zudem ein sehr positiver Aspekt.

Liegt der Fokus jedoch auf dem Aufbau einer Cassandra-Datenbank auf Azure, so bietet sich diese Plattform nur bedingt an. Negativ aufgefallen sind hierbei vor allem die veraltete Anleitung und das manuelle, und damit zeitaufwendige Installieren von Cassandra.

Die Cloud-Plattform von Microsoft kann daher nicht für einen unkomplizierten und schnellen Aufbau der NoSQL-Datenbank Cassandra empfohlen werden.

5.2 Zweiter Versuch: Cassandra auf Google Cloud

Wie bereits im ersten Versuch mit Microsoft Azure, wird im Folgenden zuerst der Cloud-Provider, Google Cloud, sowie dessen Plattform vorgestellt. Da die benötigten Hilfsprogramme den Programmen des letzten Versuches entsprechen, wird auf diese nicht erneut eingegangen. Einzige Ausnahme bildet das Programm PuTTY, welches aufgrund der in Google Cloud integrierten Konsole nicht weiter benötigt wird. Im Anschluss an die Vorstellung des Cloud-Providers wird daher direkt der Verlauf des Aufbaus von Cassandra auf der Google Cloud dokumentiert, sowie das Ergebnis vorgestellt. Ein Fazit zum Vorgang rundet das Kapitel ab.

5.2.1 Der Cloud Provider Google Cloud

Die Google Cloud ist, wie der Name bereits andeutet, die Cloud Platform von Google. Sie nutzt dieselbe Infrastruktur, mit der Google Suchergebnisse in Sekundenbruchteilen ausgibt, YouTube-Videoinhalte abspielt und Speicherplatz von Millionen Gmail-Nutzern bereitstellt.

Durch die Unterstützung von Google in der Verwaltung von App-, Datenbank- und Speicherservern, kann sich auf das Wesentliche konzentriert werden – das Entwickeln. Hierzu gehören die Datenbank Administration, Server Konfiguration und die automatische Lastenverteilung.

Auch eine Integration mit bekannten Entwicklungstools, wie zum Beispiel Eclipse, ist vorhanden. Zudem bietet Google Cloud ein Verwaltungsportal, indem die Performance von Anwendungen und das Verwalten des Accounts sowie der Rechnungen verwaltet werden kann.

Auf der Cloud Platform gehostete Anwendungen können automatisch für anspruchsvolle Arbeitslasten hochskaliert und bei nachlassendem Traffic wieder herunterskaliert werden. Gezahlt wird hierbei nur für die tatsächliche Nutzung.

Die Google Cloud bietet zudem ein breites Angebot an verfügbaren Diensten. Hierzu zählen: Virtuelle Maschinen, verwaltete Plattform, Blob-Storage, Block Storage, NoSQL-Datenspeicher, MySQL-Datenbank und Big-Data-Analysen.

Die Computing Infrastruktur von Google bietet eine konsistente CPU-, Speicher- und Festplattenleistung, sodass Latenzzeiten auf ein Minimum reduziert werden.

Die Google Cloud Platform bietet unterschiedliche Formen des Supports. Zum einen gibt es den klassischen Telefonsupport, zum anderen jedoch auch einen kostenlosen Community-basierten Support.[38]

5.2.2 Aufbau von Cassandra auf Google Cloud

Der Aufbau einer Cassandra-Datenbank auf Google Cloud wird durch ein „Click-to-Deploy" Softwarepaket vereinfacht. Nach Eingabe der relevanten Informationen (zum Beispiel der Standort des Servers, die Anzahl der Cassandra-Knoten, die Eigenschaften der virtuellen Maschinen, etc.) wird das Cassandra-Cluster nach nur wenigen Klicks automatisch erstellt. Die manuelle Installation über die Konsole entfällt hier.

Die nun angelegten virtuellen Maschinen können im Verwaltungsportal eingesehen werden. In Anhang 5 ist ein Ausschnitt des Verwaltungsportales zu sehen. Hier kann die CPU-Auslastung, der Netzwerk-Traffic und der Laufwerk-Traffic graphisch überwacht werden.

Über den Button „SSH" gelangt man in die bereits angesprochene Konsole von Google Cloud:

Abb. 8: Google Cloud Konsole

Wie man bereits an der letzten Zeile erkennen kann, werden hierüber die JavaScript-Dateien mithilfe von Node.js ausgeführt.

Da die Datenbank momentan noch keine Datensätze enthält, müssen nun per Skript Datensätze eingelesen werden. Die Reihenfolge für das erstmalige einlesen von Datensätzen läuft folgendermaßen ab:

[38] Vgl. Google (o.J.)

Abb. 9: Einlesen von Daten in Cassandra[39]

Damit Daten erstmalig eingelesen werden können, muss zuvor ein „Keyspace“ und eine „Columnfamily“ erstellt werden. Die Columnfamily ist vergleichbar mit einer Tabelle aus relationalen Datenbanken. In einem Keyspace können mehrere dieser Columnfamilies erstellt werden. Wie auch in relationalen Datenbanken werden in der Columnfamily die Spalten der Datensätze festgelegt. Spalten werden hierbei über einen frei definierbaren Namen und einen Datentyp definiert. Wurde die Columnfamily den einzulesenden Datensätzen entsprechend erstellt, können diese daraufhin eingelesen werden.

Da im ersten Versuch bereits ein funktionsfähiges Skript erstellt wurde, wird das Einlesen auch dieses Mal über ein Skript in der Skriptsprache JavaScript erfolgen. Hierfür wurde, wie bereits angesprochen, Node.js auf dem Cassandra-Server installiert. Das gesamte Skript kann dem Anhang 6 entnommen werden.

Abb. 10: Skript Teil 1

Im ersten Teil des Skriptes werden funktionsübergreifende Variablen definiert, sowie die nachfolgende Funktion „createKeyspace()“ aufgerufen. Die Variable „pooledCon“ dient hierbei später dem Verbindungsaufbau einer „Pooled Connection“, welche durch das Einbinden des „cassandra-client“-Paketes ermöglicht wird. Die Variablen „ksName“ und

[39] Vgl. Microsoft Corporation (o.J.a)

„cfName“ definieren die Bezeichnung für den anzulegenden Keyspace und die Columnfamily. Mithilfe von „ksConOptions“ wird die IP-Adresse der virtuellen Maschine sowie der Keyspace angegeben, auf der später die Columnfamily erstellt und die Daten eingelesen werden sollen.

Abb. 11: Skript Teil 2

Die Funktion „createKeyspace()“ dient, wie der Name bereits vermuten lässt, der Erstellung des Keyspace. In der Variable „cql“ ist der hierzu nötige „Cassandra Query Language (CQL)“-Befehl hinterlegt. Da der Verbindungsaufbau mithilfe der Variablen „ksConOptions“ aus Teil eins des Skriptes momentan noch nicht möglich ist, da der Keyspace „kunden“ noch nicht angelegt ist, muss zum „System-Keyspace“ verbunden werden.

Der CQL-Befehl gelangt durch den Aufruf von „con.execute(cql)“ an den Server und wird dort ausgeführt. Funktioniert dies fehlerfrei, so wird per „callback()“ die nächste Funktion, die Funktion „createColumnfamily()“, ausgeführt.

Abb. 12: Skript Teil 3

Wie auch hier bereits der Name die Funktionalität beschreibt, dient diese Funktion der Erstellung einer Columnfamily. Der dazugehörige CQL-Befehl wird wie bereits in der vorhergegangenen Funktion in der Variable „cql“ gespeichert. Wie man sehen kann enthält der Befehl oft das Zeichen „?“. Der Grund hierfür ist die hierbei verwendete Abspaltung der benötigten Informationen vom auszuführenden CQL-Befehl. Jedes „?“ steht hierbei für einen

String des Arrays „params". Deutlich wird dies beim Aufruf der Funktion „con.execute(cql, params)", bei der sowohl der CQL-Befehl als auch die benötigten Informationen übergeben werden. Die Informationen bestehen hierbei zum einen aus dem Namen der Columnfamily, „daten", sowie zum anderen aus den Spaltenbezeichnungen und deren Datentypen. Wichtig ist zudem, dass der Verbindungsaufbau nun über den Keyspace „kunden" verläuft, in dem die Columnfamily erstellt werden soll.

Verläuft das Erstellen der Columnfamily ohne einen Fehler, so wird nun erneut per „callback()" die nächste Funktion aufgerufen.

Abb. 13: Skript Teil 4

Die in der obigen Abbildung gezeigte Funktion „populateCustomerData()" dient dem Einlesen der csv-Datei mit den Datensätzen. In der Variable „filename" wird hierzu der Zielort und der Name der Datei hinterlegt.

Mithilfe des Paketes „readline" wird zu Anfang ein Einlesestream für diese Datei erzeugt. Damit die einzelnen Daten den richtigen Spalten in der Datenbank zugeteilt werden können, wird daraufhin jede eingelesene Zeile den Spalten entsprechend gesplittet und in einem Array abgelegt. Gesplittet wird hierbei durch das Abtrennen der kommagetrennten Daten. Da in der csv-Datei der einzulesenden Datensätze die Daten jeweils mit Anführungszeichen markiert sind, die Cassandra-Datenbank damit jedoch nicht umgehen kann, werden diese Zeichen durch die dargestellte Schleife entfernt. Sie entfernt das erste und letzte Zeichen eines jeden Datenstrings im Array, was den Anführungszeichen entspricht. Die Zeile ist damit zum Einfügen in die Datenbank bereit.

Mithilfe des Aufrufes „updateCustomer(ksConOptions, params)", wird die Funktion zum Schreiben der Datensätze in die Datenbank ausgeführt, und die Zeile die geschrieben werden soll übergeben. Die Funktion wird damit für jede Zeile die geschrieben werden muss neu aufgerufen.

Abb. 14: Skript Teil 5

Wie bereits in der Funktion zur Erstellung der Columnfamily wird auch hier wieder der CQL-Befehl in der Variable „cql“ abgelegt. Die einzufügenden Daten stammen hierbei aus dem übergebenen Array der Funktion zum Einlesen der Datei. Hier muss beachtet werden, dass die Arrayfelder den richtigen Spalten der Columnfamily zugeordnet werden.

Wurde ein Datensatz erfolgreich in die Datenbank geschrieben, so erfolgt eine Meldung mit einer Information über die „persnr“ des geschriebenen Datensatzes.

Soll nun überprüft werden, ob die Zuordnung der Daten zu den richtigen Spalten erfolgreich war, so lässt sich dies durch einen beliebigen Select realisieren. Das Ergebnis für die in dieser Projektarbeit verwendeten Datensätze sieht wie folgt aus:

Results | Query Trace

persnr	geb	hnr	nname	stadt	strasse	vname	vnr
72288	08.05.1934	31	Kubik	Wuppertal	Niege Reeg	Adelaide	73398
66880	17.03.2004	59	Kischel	Halle	Gendorfer Stra...	Adela	67990
110294	11.06.1969	38	Urbainczyk	Buxtehude	Bleichen	Adora	111404
116204	10.07.2009	6	Drefahl / Drep...	Halver	Emstaler Stra�e	Adorne	117314
67492	24.10.1994	46	Peukes	Creuzburg	Rehlinger Weg	Adela	68602
77542	16.07.1958	54	Krullmann	Sendenhorst	Sachsendorfer...	Adelheid	78652
6108	01.11.1989	77	Her(r)mann	Herborn	Violenstra�e	Abagael	7218
33734	17.11.1974	50	Sobe	Schwedt/Oder	Lambrechtsha...	Abigale	34844
121420	22.10.1960	71	Dick	Enger	Voeder Stra�e	Adria	122530
33747	12.08.2004	39	Sommer	Altenau	Voigendorf	Abigale	34857
105682	21.08.1981	31	Druckrey	Senftenberg	Steggasse	Adina	106792
44725	03.02.1921	9	Z�lfl	Bad Wimpfen	Hans-Ernst-Str...	Adaline	45835
72857	12.02.2010	2	Pung	Alpirsbach	Hans-Krieg-St...	Adelaide	73967
94130	22.06.1956	8	Schey	Altenberg	Ruppersdorf	Adelle	95240

Abb. 15: Ergebnis eines beliebigen Selects

Wie aus der Abbildung ersichtlich wird, hat die Zuordnung fehlerfrei funktioniert. Zur besseren Übersicht wurde für den Screenshot das Hilfsprogramm „DataStax DevCenter“ anstatt der Konsole verwendet.

5.2.3 Fazit

Google Cloud bietet eine sehr gute Unterstützung beim Aufbau einer Cassandra-Datenbank und punktet ebenfalls mit einem sehr übersichtlichen Verwaltungsportal. Die Dokumentationen sind auf dem neuesten Stand und verweisen oft auf sehr hilfreiche, weiterführende Informationsseiten.

Durch eine gute Integration der DataStax-Produkte sowie weiterer Eclipse-basierter Unterstützungstools wird die Handhabung auch für NoSQL-Neulinge vereinfacht.

Insgesamt bietet Google Cloud damit einige Vorteile gegenüber Microsoft Azure, was den Aufbau und Betrieb einer Cassandra-Datenbank betrifft.

6 Testszenarien

Im Folgenden soll anhand von acht Testszenarien die Praxistauglichkeit der beiden vorgestellten NoSQL-Datenbanksysteme untersucht werden. Dazu wurden typische Anwendungsfälle relationaler Datenbanksysteme ermittelt und in den Systemen DynamoDB und Cassandra angewandt. Ein Testszenario besteht aus einem SQL-Kommando und je einer Umsetzung zusammen mit einem Codebeispiel mit DynamoDB und Cassandra. Für jedes Szenario wurde, falls möglich, die durchschnittlich benötigte Ausführungszeit berechnet. Diese wurde jeweils durch 25-malige Wiederholung des Szenarios ermittelt.

Zur Durchführung der Szenarien standen DynamoDB 1000 Lese- und 25 Schreibkapazitätseinheiten zur Verfügung.

6.1 Ausgabe aller gespeicherten Daten

SQL-Kommando:

```
SELECT * FROM kunden;
```

Umsetzung mit DynamoDB:

Für DynamoDB wird die Scan-Operation (siehe *Tab. 3*) benötigt, um alle Elemente aus der Tabelle auslesen zu können. Es ist nicht möglich, mit einer einzigen Operation alle gespeicherten Elemente abzurufen, da das Ergebnis einer Scan-Operation einer Größenbeschränkung von 1 MB unterliegt. Teil eines Scan-Ergebnisses ist der sogenannte *LastEvaluatedKey.*[40] Ist der *LastEvaluatedKey* gesetzt bedeutet dies, dass noch weitere Datensätze abgefragt werden können. Um diese weiteren Datensätze abzufragen, wird derselbe Scan-Request zusammen mit dem *LastEvaluatedKey* nochmals abgeschickt. Dies wird so oft wiederholt, bis der *LastEvaluatedKey* leer ist, d. h. dass keine weiteren Datensätze abzufragen sind (siehe Abb. 16).

[40] Vgl. Amazon Web Services (2014c)

```
Map<String, AttributeValue> lastPrimaryKey = null;

do {
    ScanRequest scanRequest = new ScanRequest()
        .withTableName(tableName)
        .withExclusiveStartKey(lastPrimaryKey);

    ScanResult result = client.scan(scanRequest);

    for (Map<String, AttributeValue> item : result.getItems()) {
        System.out.println(item);
    }

    lastPrimaryKey = result.getLastEvaluatedKey();
} while (lastPrimaryKey != null);
```

Abb. 16: Implementierung von Testszenario 1 mit DynamoDB

Da hier mit der Scan-Operation gearbeitet werden muss, war es nicht möglich die tatsächlich benötigte Zeit, um alle Werte auszulesen, zu ermitteln.

Durchschnittlich benötigte Zeit, um 100.000 Datensätze auszulesen: 31.543,3ms

Umsetzung mit Cassandra:

In Cassandra kann der SQL-Befehl wie oben angegeben übernommen werden. Der einzige Unterschied liegt im FROM-Teil, da hier die Informationen des Keyspace und der Columnfamily angegeben werden müssen. Für die Syntax bedeutet dies:

„SELECT * FROM kunden.daten;“

Der Keyspace wird also vor der Columnfamily angegeben und mit einem Punkt getrennt. Da das von uns verwendete Tool zur Übermittlung der CQL-Befehle, „DataStax DevCenter“, jedoch bei der Abfrage von mehr als 5.000 Zeilen einen Fehler wirft, muss der Umweg über eine JavaScript-Datei erfolgen. Hier wird lediglich der CQL-Befehl wie oben angegeben übergeben, ausgeführt und das Ergebnis zurückgegeben.

6.2 Ausgabe des Datensatzes mit der Personennummer 1000

SQL-Kommando:

```
SELECT * FROM kunden WHERE persnr = 1000;
```

Umsetzung mit DynamoDB:

Um einen Datensatz mit einem bestimmten Primärschlüssel zu lesen, wird die GetItem-Operation benötigt. Die Verwendung ist in *Abb. 17* abgebildet.

```
Table table = dynamoDB.getTable(tableName);

Item item = table.getItem("persnr", 1000);

System.out.println(item);
```

Abb. 17: Implementierung von Testszenario 2 mit DynamoDB

Durchschnittlich benötigte Zeit: 411.2ms

Umsetzung mit Cassandra:

Da bei dieser Abfrage nur 1 Datensatz verlangt wird, kann die Operation mit dem DataStax DevCenter durchgeführt werden. Hierbei wird der SQL-Befehl wie bereits im ersten Testszenario um die Angabe des Keyspaces und der Columnfamily ergänzt und ausgeführt. Das Ergebnis entspricht dem erwarteten Ergebnis: Der Datensatz mit der Personennummer 1000 wird angezeigt.

Durchschnittlich benötigte Zeit: 881.3ms

6.3 Alle Versicherungsnummern von Kunden aus Leipzig ausgeben

SQL-Kommando:

```
SELECT vnr FROM kunden WHERE stadt = 'Leipzig';
```

Umsetzung mit DynamoDB:

```
Condition stadtCondition = new Condition()
    .withComparisonOperator(ComparisonOperator.EQ)
    .withAttributeValueList(new AttributeValue().withS("Leipzig"));

Map<String, Condition> conditions = new HashMap<>();
conditions.put("stadt", stadtCondition);

Map<String, AttributeValue> lastPrimaryKey = null;

do {
    ScanRequest scanRequest = new ScanRequest()
        .withTableName(tableName)
        .withScanFilter(conditions)
        .withExclusiveStartKey(lastPrimaryKey);

    ScanResult result = client.scan(scanRequest);
    for (Map<String, AttributeValue> item : result.getItems()) {
        System.out.println(item);
    }

    lastPrimaryKey = result.getLastEvaluatedKey();
} while (lastPrimaryKey != null);
```

Abb. 18: Implementierung von Testszenario 3 mit DynamoDB

Da in der Tabelle *Kunden* nur ein Primärschlüssel für die Personennummer definiert ist, muss mit einer Scan-Operation über die gesamte Tabelle gearbeitet werden. Die Implementierung (siehe *Abb. 18*) ist nahezu identisch mit der in *Abb. 16* dargestellten, wurde

jedoch durch eine Bedingung erweitert. Die DynamoDB API kennt folgende Vergleichsoperatoren: EQ | NE | IN | LE | LT | GE | GT | BETWEEN | NOT_NULL | NULL | CONTAINS | NOT_CONTAINS | BEGINS_WITH[41]. Um nur Kunden aus Leipzig auszugeben wird der Operator EQ gewählt, mit dessen Hilfe geprüft wird, ob der Wert des Attributs stadt mit der Zeichenkette „Leipzig“ übereinstimmt.

Da in diesem Szenario ebenso wie in Testszenario 1 mit der Scan-Operation gearbeitet werden muss, war es nicht möglich, den tatsächlichen Zeitverbrauch für die Abfrage aller Kunden aus Leipzig zu ermitteln.
Durchschnittlich benötigte Zeit, um 100.000 Kunden aus Leipzig auszulesen: 333,651.1ms

Umsetzung mit Cassandra:

Um die Umsetzung des Testszenarios in Cassandra durchführen zu können, muss zuerst ein Index über die Spalte „stadt“ erzeugt werden. Dies erfolgt mithilfe des Befehls:

„CREATE INDEX stadt ON kunden.daten (stadt);“

Wurde der Index erfolgreich erzeugt, kann nun der SQL-Befehl über das DevCenter auf der Datenbank ausgeführt werden.

Durchschnittlich benötigte Zeit: 7,942.2ms

6.4 Einfügen eines Datensatzes

Der neue Datensatz soll folgende Werte besitzen:

Attribut	Wert
persnr	123456
nname	Kohl
vname	Helmut Josef Michael
geb	03.04.1930
stadt	Ludwigshafen am Rhein
straße	Hauptstraße
hnr	11
vnr	98765

Tab. 4: Attribute des neuen Datensatzes für Testszenario 4

[41] Vgl. Amazon Web Services (2012)

SQL-Kommando:

```
INSERT INTO kunden VALUES (123456, 'Kohl', 'Helmut Josef Michael',
'03.04.1930', 'Ludwigshafen am Rhein', 'Hauptstraße', 11, 98765);
```

Umsetzung mit DynamoDB:

Um ein Element in eine Tabelle der DynamoDB einzufügen, kann wie in Abb. 19 vorgegangen werden: es wird ein strukturiertes *Item* mit den gewünschten Werten des neuen Eintrags angelegt und mit der *PutItem*-Operation in die Tabelle geschrieben.

```
Table table = dynamoDB.getTable(tableName);

Item item = new Item()
    .withPrimaryKey("persnr", 123456)
    .withString("nname", "Kohl")
    .withString("vname", "Helmut Josef Michael")
    .withString("geb", "03.04.1930")
    .withString("stadt", "Ludwigshafen am Rhein")
    .withString("straße", "Hauptstraße")
    .withNumber("hnr", 11)
    .withString("vnr", "98765");

table.putItem(item);
```

Abb. 19: Implementierung von Testszenario 4 mit DynamoDB

Durchschnittlich benötigte Zeit: 409.7ms

Umsetzung mit Cassandra:

Das Einfügen eines Datensatzes in die Cassandra-Datenbank erfolgt anhand einer Zuweisung der einzufügenden Daten mit den Spalten. Der CQL-Befehl lautet daher:

„INSERT INTO kunden.daten (persnr, nname, vname, geb, stadt, strasse, hnr, vnr) VALUES (123456, ‚Kohl‘, ‚Helmut Josef Michael‘, ‚03.04.1930‘, ‚Ludwigshafen am Rhein‘, ‚Hauptstraße‘, 11, 98765);“

Durchschnittlich benötigte Zeit: 937.5ms

6.5 Einfügen eines Datensatzes ohne Angabe des Primärschlüssels

Der neue Datensatz soll folgende Werte besitzen

Attribut	Wert
nname	Adenauer
vname	Konrad Hermann Joseph
geb	05.01.1876

Tab. 5: Attribute des neuen Datensatzes für Testszenario 5

SQL-Kommando:

```
INSERT INTO kunden (nachname, vorname, geburtsdatum) VALUES ('Adenauer', 'Konrad Hermann Joseph', '05.01.1876');
```

Umsetzung mit DynamoDB:

Dieses Szenario ist auf die DynamoDB nicht anwendbar, da für jedes Element, das in eine Tabelle eingefügt werden soll, ein Primärschlüssel angegeben sein muss. Ein automatisch hochzählender Primärschlüssel wird nicht unterstützt.

Umsetzung mit Cassandra:

Das Einfügen dieses Datensatzes kann in Cassandra nicht erfolgen. Wie auch in der DynamoDB können hier keine Datensätze eingefügt werden, die keine Information über den Primärschlüssel enthalten.

6.6 Aktualisierung eines Datensatzes

Für den Datensatz mit der Personennummer 1237901 sollen folgende Attribute geändert werden:

Attribut	Wert
stadt	Regensburg
straße	Bahnhofstraße
hnr	101

Tab. 6: Neue Werte des betroffenen Datensatzes in Testszenario 6

SQL-Kommando:

```
UPDATE kunden SET stadt = 'Regensburg', straße = 'Bahnhofstraße',
hausnummer = 101 WHERE persnr = 1237901;
```

Umsetzung mit DynamoDB:

Um Attribute von Items zu ändern, wird die *UpdateItem*-Operation benötigt (siehe Abb. 20). Dazu werden die zu ändernden Attribute mit den neuen Werten in einer Liste angegeben. Zusammen mit dem Primärschlüssel des Items ergibt dies die *UpdateItemSpec*, die angibt welches Item wie geändert werden soll.

```
Table table = dynamoDB.getTable(tableName);

List<AttributeUpdate> attributeUpdates = new LinkedList<>();
attributeUpdates.add(new AttributeUpdate("stadt").put("Regensburg"));
attributeUpdates.add(new AttributeUpdate("straße").put("Bahnhofstraße"));
attributeUpdates.add(new AttributeUpdate("hnr").put(101));

UpdateItemSpec spec = new UpdateItemSpec()
    .withPrimaryKey("persnr", 1237901)
    .withAttributeUpdate(attributeUpdates);

table.updateItem(spec);
```

Abb. 20: Implementierung von Testszenario 6 mit DynamoDB

Durchschnittlich benötigte Zeit: 466.3ms

Umsetzung mit Cassandra:

Zum Updaten eines Datensatzes reicht in Cassandra wie zuvor ein kurzer CQL-Befehl aus:

„UPDATE kunden.daten SET stadt = ‚Regensburg', strasse = ‚Bahnhofstraße', hnr = 101 WHERE persnr = 1237901;"

Durchschnittlich benötigte Zeit: 922.1ms

6.7 Löschen eines Datensatzes

Es soll der Datensatz mit der Personennummer 1000 aus der Tabelle gelöscht werden.

SQL-Kommando:

```
DELETE FROM kunden WHERE persnr = 1000;
```

Umsetzung mit DynamoDB:

Das Löschen eines Items erfolgt mit der *DeleteItem*-Operation. Das Item wird über dessen Primärschlüssel identifiziert und aus der Tabelle entfernt (siehe Abb. 21).

```
Table table = dynamoDB.getTable(tableName);

table.deleteItem("persnr", 1000);
```

Abb. 21: Implementierung von Testszenario 7 mit DynamoDB

Durchschnittlich benötigte Zeit: 453.5ms

Umsetzung mit Cassandra:

Der Datensatz mit der Personennummer 1000 wird mittels dem Befehl:

„DELETE FROM kunden.daten WHERE persnr = 1000;"

gelöscht.

Durchschnittlich benötigte Zeit: 906.3ms

6.8 Löschen von mehreren Datensätzen

Es sollen Datensätze mit den Personennummern 2500, 11517356 und 22469735 entfernt werden.

SQL-Kommando:

```
DELETE FROM kunden WHERE persnr IN (2500, 11517356, 22469735);
```

Umsetzung mit DynamoDB:

Mehrere Items können durch die mehrfache Verwendung der *DeleteItem*-Operation aus der Tabelle entfernt werden (siehe *Abb. 22*)

```
Table table = dynamoDB.getTable(tableName);

int[] primaryKeys = new int[] { 2500, 11517356, 22469735 };

for (int pk : primaryKeys) {
    table.deleteItem("persnr", pk);
}
```

Abb. 22: Implementierung von Testszenario 8 mit DynamoDB

Durchschnittlich benötigte Zeit: 553.7ms

Umsetzung mit Cassandra:

Das Löschen von mehreren Datensätzen über den Primärschlüssel wird in Cassandra mit dem IN-Operator umgesetzt:

„DELETE FROM kunden.daten WHERE persnr IN (2500, 11517356, 22469735);“

Durchschnittlich benötigte Zeit: 937.6ms

7 Fazit und Ausblick

Wie in der Aufgabenstellung formuliert, war das Ziel dieser wissenschaftlichen Arbeit der Aufbau von NoSQL-Datenbanken in der Cloud und damit verbunden die Prüfung der Praxistauglichkeit.

Im Rahmen dieser Arbeit wurde eine DynamoDB auf der Cloud des Providers AWS implementiert und eine Cassandra-Datenbank auf der Cloud von Google.

Cassandra ist eine sehr SQL-nahe Datenbank, deren Einrichtung und Verwendung unkompliziert möglich ist. Die Nutzung der Cloud von Google stellte sich ebenso als Erfolg heraus. Google bietet ein benutzerfreundliches Verwaltungsportal und zudem ein umfangreiches Dokumentationswerk. Der Aufbau von Cassandra auf Microsoft Azure ist dagegen aufgrund mangelhafter Dokumentationen misslungen. Momentan ist daher davon abzuraten, Azure als Cloud-Betreiber einer Cassandra-Datenbank zu nutzen.

DynamoDB ließ sich ebenso unkompliziert auf der Cloud von bei AWS einrichten. Hier ist jedoch festzuhalten, dass etwaige Indexe während Erstellung der Datenbank eingerichtet werden müssen. Ein nachträgliches Hinzufügen ist momentan nicht vorgesehen, ist jedoch angedacht.

Um Testszenarien an beiden Datenbanken zu ermöglichen, wurde eine Datenbank aufgebaut. Dies liegt zum einen an der knappen Verfügbarkeit von Massendaten und zum anderen an der mangelnden Vergleichbarkeit mit versicherungstypischen Datensätzen. Aufgrund dessen war die Beschaffung geeigneter offener Daten im Zeitrahmen dieses Projektes nicht möglich. Stattdessen wurde mittels Java-Code eine eigene Testdatenbank erzeugt. Diese umfasst rund 23 Millionen Kundendatensätze.

Nach Aufbau der Datenbanken und Erstellung der Testdatenbank erfolgten einige in ihrer Komplexität gesteigerte Testdurchläufe. Hierbei zeigte die DynamoDB Schwächen bei der abrufbaren Datenmenge pro Select. Aufgrund von Beschränkungen muss eine Abfrage bei großen Datenmengen mehrmals gestartet werden. Sowohl bei Cassandra als auch DynamoDB war es zudem nicht möglich Datensätze ohne Primärschlüssel einzufügen.

Nach eingehender Beschäftigung mit NoSQL-Datenbanken wurde deutlich, dass diese schwer mit klassischen RDMS zu vergleichen sind. NoSQL-Datenbanken haben andere Stärken, wie beispielsweise eine hohe Skalierbarkeit und einen einfachen Zugriff. Daher können sie in gewissen Bereichen als Alternative für klassische Modelle genutzt werden, vollständig ersetzen, können Sie diese Stand heute aber noch nicht.

Inwieweit sich komplexere Datenbanksysteme auf die Funktionalitäten von NoSQL-Datenbanken auswirken, konnte im Rahmen dieser wissenschaftlichen Arbeit nicht

eingehend untersucht werden. Diese Thematik eignet sich daher, neben der eindringlicheren Befassung mit Open Data, für weitere Studien.

Anhang

Anhang 1: Quellcode: Erzeugung von Testdatensätzen

```
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.GregorianCalendar;
import java.util.List;
import java.util.Random;

import com.opencsv.CSVReader;
import com.opencsv.CSVWriter;

public class Generator {

	/**
	 * @param args
	 */

	int pNr = 1;
	String vName;
	String nName;
	Date geburt;
	String ort;
	String straße;
	int hausnummer;
	int vNr = 1111;

	String strPnr = "";
	String strVnr = "";
	String strHausnummer = "";
	String strGeburt = "";

	// CSV-Dateien
	String csvVname = "/Users/Johannes/Documents/Studium/DHBW/5.
Semester/Projekt/CSV/Vornamen.csv";
	String csvNname = "/Users/Johannes/Documents/Studium/DHBW/5.
Semester/Projekt/CSV/Nachnamen.csv";
	String csvStraße = "/Users/Johannes/Documents/Studium/DHBW/5.
Semester/Projekt/CSV/Straßen.csv";
	String csvOrt = "/Users/Johannes/Documents/Studium/DHBW/5.
Semester/Projekt/CSV/Städte.csv";
	String csvFinal = "/Users/Johannes/Documents/Studium/DHBW/5.
Semester/Projekt/CSV/Dat10.csv";

	// Listen mit Daten
	static ArrayList<String> straßen = new ArrayList<String>();
	static ArrayList<String> orte = new ArrayList<String>();

	// MAin-Methode
	public static void main(String[] args) throws IOException {

		Generator gen = new Generator();
		straßen = gen.getStreetData();
		orte = gen.getOrtData();
```

```
		gen.create();
	}

	// erstellt neue Datei als Kombination der Daten aus den anderen Dateien
	private void create() {

		try {

			// zu schreibende Datei
			CSVWriter file = new CSVWriter(new FileWriter(csvFinal));

			CSVReader brVname = new CSVReader(new FileReader(csvVname));
			String[] rowVname;

			// verbindet jeden Vornamen mit jedem Nachnamen
			// einlesen jedes Vornamen
			while ((rowVname = brVname.readNext()) != null) {
				vName = rowVname[0];

				CSVReader brNname = new CSVReader(new
FileReader(csvNname));
				String[] rowNname;
				while ((rowNname = brNname.readNext()) != null) {
					nName = rowNname[0];

					// Werte für den Datensatz werden berechnet
					geburt = getGeburtsdatum();
					ort = getOrt();
					straße = getStraße();
					hausnummer = getHausnummer();

					// wandelt INteger in String um
					strPnr = strPnr.valueOf(pNr);
					strVnr = strVnr.valueOf(vNr);
					strHausnummer = strHausnummer.valueOf(hausnummer);

					// formatiert das Geburtsdatum
					SimpleDateFormat form = new
SimpleDateFormat("dd.MM.yyyy");
					strGeburt = form.format(geburt);

					// schreibt in finale Datei
					List<String[]> data = new ArrayList<String[]>();
					data.add(new String[] { strPnr, nName, vName,
strGeburt,
							ort, straße, strHausnummer, strVnr
});
					file.writeAll(data);

					pNr++;
					vNr++;

				}
				brNname.close();

			}
			brVname.close();

			file.close();

		} catch (FileNotFoundException e) {
```

```
				// TODO Auto-generated catch block
				e.printStackTrace();
			} catch (IOException e) {
				// TODO Auto-generated catch block
				e.printStackTrace();
			}

	}

	private ArrayList getStreetData() throws IOException {
		// Straßenliste
		CSVReader brStraße = new CSVReader(new FileReader(csvStraße));
		String[] rowStraße;

		ArrayList<String> straßen = new ArrayList<String>();

		while ((rowStraße = brStraße.readNext()) != null) {
			straßen.add(rowStraße[0]);
		}

		brStraße.close();

		return straßen;

	}

	private ArrayList getOrtData() throws IOException {
		CSVReader brOrt = new CSVReader(new FileReader(csvOrt));
		String[] rowOrt;

		ArrayList<String> orte = new ArrayList<String>();

		while ((rowOrt = brOrt.readNext()) != null) {
			orte.add(rowOrt[0]);
		}
		brOrt.close();

		return orte;
	}

	// wählt eine zufällige Hausnummer zwischen 1 und 80
	private int getHausnummer() {

		int nr = 0;
		Random rand = new Random();
		nr = rand.nextInt(80 - 2) + 1;

		return nr;
	}

	// wählt eine zufällige Straße aus der Datei
	private String getStraße() throws IOException {

		Random rand = new Random();
		int zufallswert;
		String straße;

		zufallswert = rand.nextInt(straßen.size() - 1) + 1;
		straße = straßen.get(zufallswert);

		return straße;
```

```
        }

        // wählt einen zufälligen Ort aus der Datei
        private String getOrt() throws IOException {

                Random rand = new Random();
                int zufallswert;
                String ort;

                zufallswert = rand.nextInt(orte.size() - 1) + 1;
                ort = orte.get(zufallswert);

                return ort;
        }

        // wählt ein zufälliges Geburtsdatum
        private Date getGeburtsdatum() {

                GregorianCalendar cal = new GregorianCalendar();
                Random rand = new Random();
                Date dat = new Date();

                int year = rand.nextInt(2014 - 1921) + 1920;
                int month = rand.nextInt(12 - 2) + 1;
                int day = rand.nextInt(28 - 2) + 1;

                cal.set(cal.YEAR, year);
                cal.set(cal.MONTH, month);
                cal.set(cal.DAY_OF_MONTH, day);
                cal.set(cal.HOUR_OF_DAY, 0);
                cal.set(cal.MINUTE, 0);
                cal.set(cal.SECOND, 0);

                dat = cal.getTime();

                return dat;
        }

}
```

```
package de.dhbw.nosql;

import java.io.FileInputStream;
import java.io.InputStreamReader;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import au.com.bytecode.opencsv.CSVReader;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Region;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.dynamodbv2.AmazonDynamoDBClient;
import com.amazonaws.services.dynamodbv2.model.AttributeValue;
import com.amazonaws.services.dynamodbv2.model.BatchWriteItemRequest;
import com.amazonaws.services.dynamodbv2.model.BatchWriteItemResult;
import com.amazonaws.services.dynamodbv2.model.PutRequest;
import com.amazonaws.services.dynamodbv2.model.ReturnConsumedCapacity;
import com.amazonaws.services.dynamodbv2.model.WriteRequest;

public class DataImporter {

    static AmazonDynamoDBClient client = new AmazonDynamoDBClient(new ProfileCrede
ntialsProvider());

    static {
        client.setRegion(Region.getRegion(Regions.EU_CENTRAL_1));
    }

    static String tableName = "Kunden";

    public static void main(String[] args) throws Exception {

        try {
            importData();
        } catch (AmazonServiceException ase) {
            System.err.println("Data load script failed.");
        }
    }

    private static void importData() {

        try {
             Map<String, List<WriteRequest>> requestItems = new HashMap<String, Lis
t<WriteRequest>>();

            List<WriteRequest> tableRows = new ArrayList<WriteRequest>();

            BatchWriteItemResult result = null;
            BatchWriteItemRequest batchWriteItemRequest = new BatchWriteItemReques
t()
                    .withReturnConsumedCapacity(ReturnConsumedCapacity.TOTAL);
```

```
            CSVReader reader = new CSVReader(new InputStreamReader(new FileInputSt
ream("Daten.csv"), "Cp1252"));
            String[] line;

            while ((line = reader.readNext()) != null) {
                 Map<String, AttributeValue> item = new HashMap<String, AttributeVa
lue>();

                item.put("persnr", new AttributeValue().withN(line[0]));
                item.put("nname", new AttributeValue().withS(line[1]));
                item.put("vname", new AttributeValue().withS(line[2]));
                item.put("geb", new AttributeValue().withS(line[3]));
                item.put("stadt", new AttributeValue().withS(line[4]));
                item.put("straße", new AttributeValue().withS(line[5]));
                item.put("hnr", new AttributeValue().withN(line[6]));
                item.put("vnr", new AttributeValue().withS(line[7]));

                tableRows.add(new WriteRequest().withPutRequest(new PutRequest().w
ithItem(item)));

                if (tableRows.size() == 25) {
                    requestItems.put(tableName, tableRows);

                    do {
                        batchWriteItemRequest.withRequestItems(requestItems);

                        try {
                        result = client.batchWriteItem(batchWriteItemRequest);
                        // Check for unprocessed keys which could happen if you
                        // exceed provisioned throughput
                        if (!result.getUnprocessedItems().isEmpty()) {
                            System.out.println("Unprocessed              requests:
" + result.getUnprocessedItems().size());
                        }

                        requestItems = result.getUnprocessedItems();
                        } catch (Exception e) {
                            System.out.println("Current line: " + line[0]);
                            e.printStackTrace();
                        }


                    } while (result.getUnprocessedItems().size() > 0);

                    tableRows.clear();
                    requestItems.clear();
                }
            }

            // push remaining items
            if (tableRows.size() != 0) {
                System.out.println("pushing last items...");
                requestItems.put(tableName, tableRows);

                do {
                    batchWriteItemRequest.withRequestItems(requestItems);

                    try {
                    result = client.batchWriteItem(batchWriteItemRequest);
```

```
                    // Check for unprocessed keys which could happen if you
                    // exceed provisioned throughput
                    if (!result.getUnprocessedItems().isEmpty()) {
                        System.out.println("Unprocessed                  requests:
" + result.getUnprocessedItems().size());
                    }

                    requestItems = result.getUnprocessedItems();
                    } catch (Exception e) {
                        System.out.println("Current line: " + line[0]);
                        e.printStackTrace();
                    }


                } while (result.getUnprocessedItems().size() > 0);

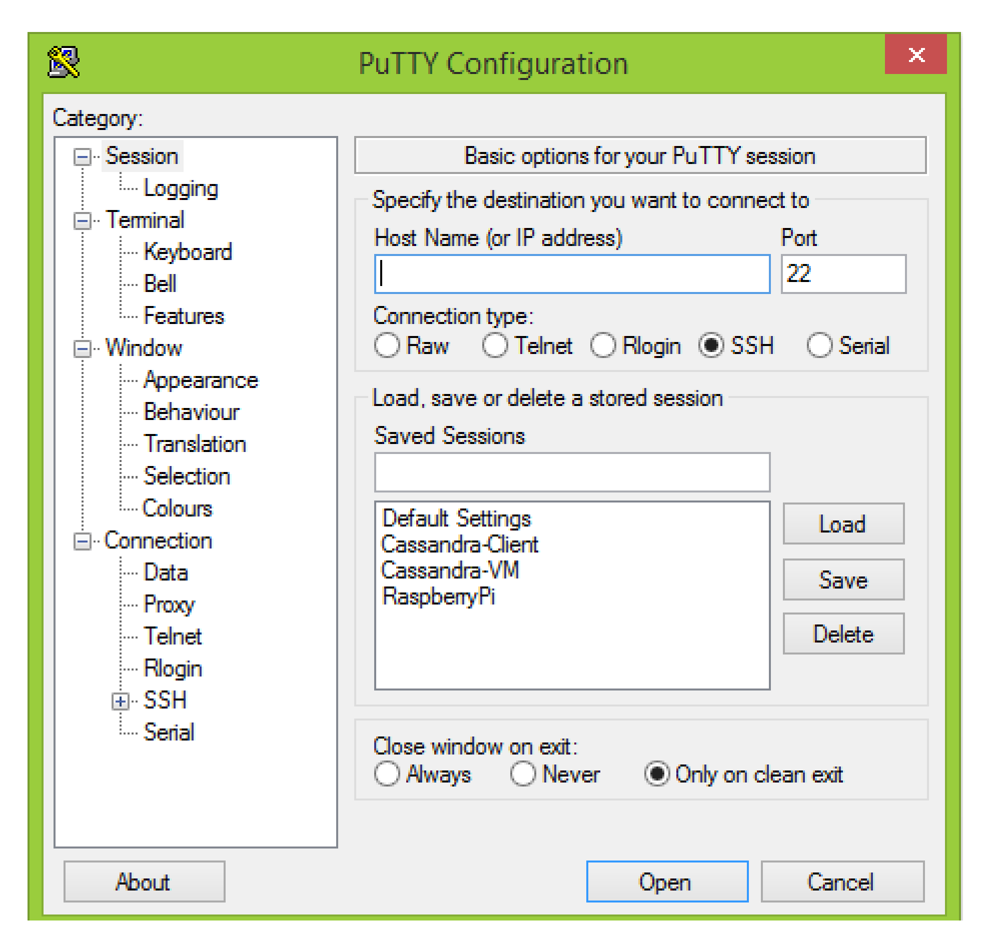
                tableRows.clear();
                requestItems.clear();
            }

        } catch (Exception e) {
            System.err.println("Failed   to   create   item   in   " + tableName + ":
" + e);
        }

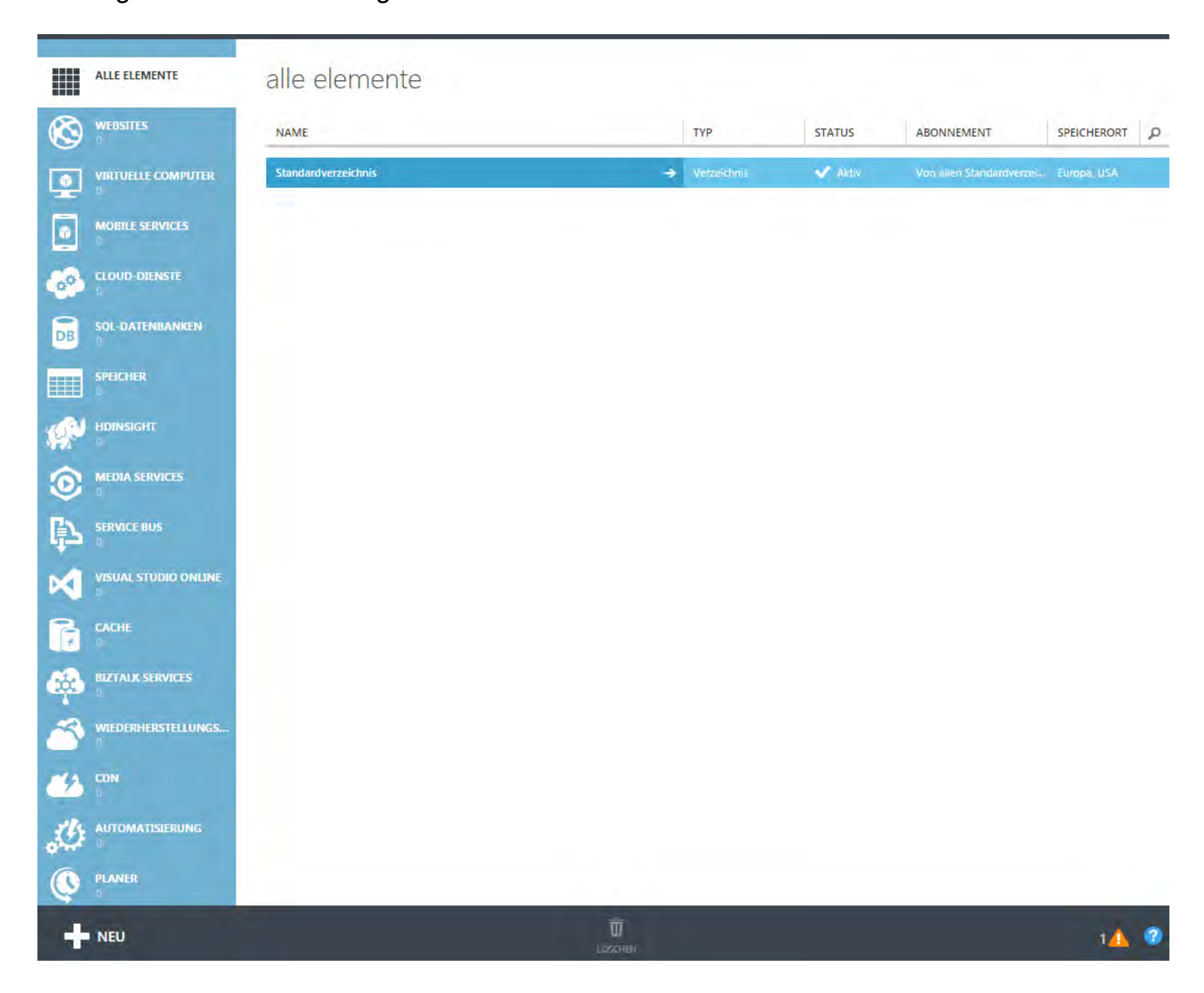
    }

}
```

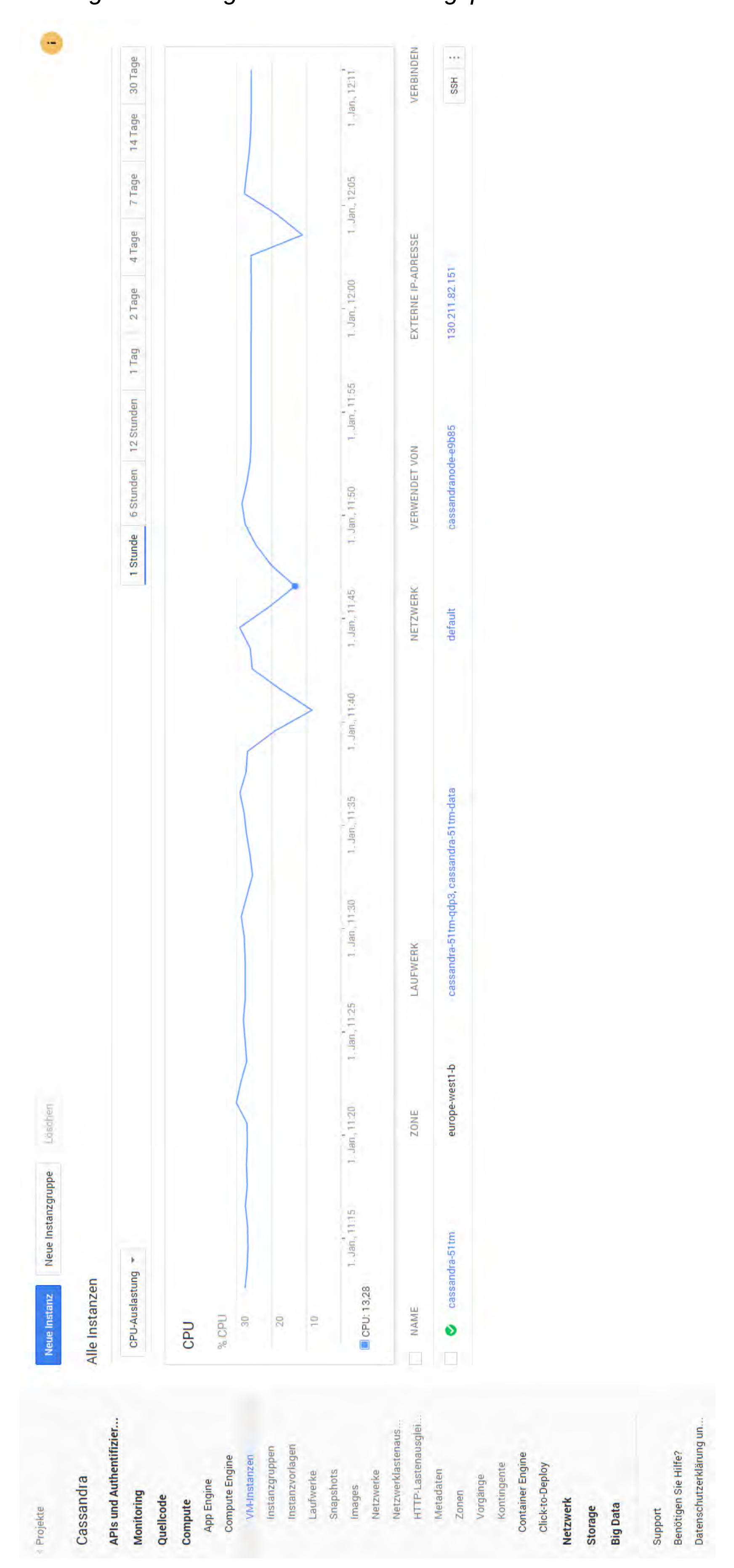
Anhang 3: PuTTY Einstiegsmaske



Anhang 4:   Azure-Einstiegsmaske



Anhang 5: Google Cloud Verwaltungsportal



Anhang 6: Skript zum Einlesen

```javascript
var pooledCon = require('cassandra-client').PooledConnection;
var ksName = "kunden";
var cfName = "daten";
var ksConOptions = { hosts: ['10.240.1.225:9160'], keyspace: ksName, use_biginits: false};

createKeyspace(createColumnFamily);

function createKeyspace(callback){
    var cql = 'CREATE KEYSPACE ' + ksName + ' WITH strategy_class=SimpleStrategy AND strategy_options:replication_factor=1';
    var sysConOptions = { hosts: ['10.240.1.225:9160'], keyspace: 'system', use_biginits: false };
    var con = new pooledCon(sysConOptions);
    con.execute(cql,[],function(err) {
    if (err) {
        console.log("Failed to create Keyspace: " + ksName);
        console.log(err);
    }
    else {
        console.log("Created Keyspace: " + ksName);
        callback(ksConOptions, populateCustomerData);
    }
    });
    con.shutdown();
}

function createColumnFamily(ksConOptions, callback){
    var params = ['daten','persnr','int','nname','text','vname','text','geb','text','stadt','text','strasse','text','hnr','int','vnr','int'];
    var cql = 'CREATE COLUMNFAMILY ? (? ? PRIMARY KEY,? ?,? ?,? ?,? ?,? ?,? ?,? ?)';
    var con = new pooledCon(ksConOptions);
    con.execute(cql,params,function(err) {
        if (err) {
            console.log("Failed to create column family: " + params[0]);
            console.log(err);
        }
        else {
            console.log("Created column family: " + params[0]);
            callback();
        }
    });
    con.shutdown();
}

function populateCustomerData() {
    var fs = require('fs');
    var filename = "./Dat10.csv";
    var contentArray;
    var params;

    require('readline').createInterface({
        input: fs.createReadStream(filename),
        terminal: false
    }).on('line', function(line){
        contentArray = line.toString().split(",");
        for (i = 0; i <= 7; i++) {
            contentArray[i] = contentArray[i].substr(1, contentArray[i].length-2);
        }
        params = [contentArray[0], contentArray[1], contentArray[2], contentArray[3], contentArray[4], contentArray[5], contentArray[6], contentArray[7]];
        updateCustomer(ksConOptions, params);
    });
}

function updateCustomer(ksConOptions, params) {
    var cql = 'INSERT INTO ' + cfName + ' (persnr, nname, vname, geb, stadt, strasse, hnr ,vnr) VALUES (' + params[0] + ', \'' + params[1] + '\', \'' + params[2] + '\', \'' + params[3] + '\', \'' + params[4] + '\', \'' + params[5] + '\', ' + params[6] + ', ' + params[7] + ')';
    var con = new pooledCon(ksConOptions);
    con.execute(cql, params, function(err) {
        if (err) console.log(err);
        else console.log("Inserted customer : " + params[0]);
    });
    con.shutdown();
}
```

Quellenverzeichnis

Verzeichnis der Internet- und Intranetquellen

Amazon Web Services (2012): Amazon DynamoDB, Condition, http://docs.aws.amazon.com/amazondynamodb/latest/APIReference/API_Condition.html, Abruf: 18.01.2015

Amazon Web Services (2014a): Improving Data Access with Secondary Indexes, http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/SecondaryIndexes.html, Abruf: 18.01.2015

Amazon Web Services (2014b): Amazon DynamoDB: Developer Guide, http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/dynamodb-dg.pdf, Abruf: 10.01.2015

Amazon Web Services (2014c): Limits in DynamoDB, http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Limits.html, Abruf: 18.01.2015

Amazon Web Services (2015a): Informationen zu AWS, http://aws.amazon.com/de/about-aws/, Abruf: 10.01.2015

Amazon Web Services (2015b): Weltweite Infrastruktur, http://aws.amazon.com/de/about-aws/global-infrastructure/, Abruf: 10.01.2015

Amazon Web Services (2015c): Produkte und Services, http://aws.amazon.com/de/products, Abruf: 10.01.2015,

Amazon Web Services (2015d): Häufig gestellte Fragen, http://aws.amazon.com/de/dynamodb/faqs, Abruf: 10.01.2015

Amazon Web Services (2015e): Amazon DynamoDB – Preise, http://aws.amazon.com/de/dynamodb/pricing/, Abruf: 10.01.2015

Amazon Web Services (2015f): Manage Users,
http://aws.amazon.com/de/iam/details/manage-users/, Abruf: 10.01.2015

Amazon Web Services (2015g): Manage Permissions and Policies,
http://aws.amazon.com/de/iam/details/manage-permissions/, Abruf: 10.01.2015

Bibliotheksverbund Bayern (2014): B3Kat – Open Data,
http://www.bib-bvb.de/web/b3kat/open-data, Abruf: 28.12.2014

Bösswetter, D. (2010): Spaltenorientierte Datenbanken,
http://www.gi.de/service/informatiklexikon/detailansicht/article/spaltenorientierte-datenbanken.html, Abruf: 13.01.2015

Dietrich, D. (2011): Was sind offene Daten?,
http://www.bpb.de/gesellschaft/medien/opendata/64055/was-sind-offene-daten?p=all, Abruf: 03.01.2015

data.gov.uk (o.J.a): Datasets,
http://data.gov.uk/data/search, Abruf: 30.12.2014

data.gov.uk (o.J.b): About,
http://data.gov.uk/about, Abruf: 30.12.2014

Finanzbehörde Hamburg (o.J.a): Herzlich Willkommen,
https://www.govdata.de/, Abruf: 30.12.2014

Finanzbehörde Hamburg (o.J.b): Daten,
https://www.govdata.de/daten, Abruf: 30.12.2014

Google (o.J.): Vorzüge der Google-Infrastruktur,
https://cloud.google.com/why-google/,
Abruf: 28.12.2014

Matzat, L. (2011): Ein Glossar rund um Open Data,
http://blog.zeit.de/open-data/2011/05/13/begriff-definition-opengov/,
Abruf: 03.01.2015

Microsoft Corporation (o.J.a): Cassandra-Linux,
https://acomdpsstorage.blob.core.windows.net/dpsmedia-prod/azure.microsoft.com/de-de/documentation/articles/virtual-machines-linux-nodejs-running-cassandra/20141112084557/cassandra-linux3.png,
Abruf: 28.12.2014

Microsoft Corporation (o.J.b): Was ist Azure?,
http://azure.microsoft.com/de-de/overview/what-is-azure/,
Abruf: 20.12.2014

OKF DE (o.J.): Was sind offene Daten?,
http://okfn.de/opendata/, Abruf: 04.01.2014

Seeger, M. (2009): Key-Value stores, a practical overview,
http://d2tyy2n2j2cu1h.cloudfront.net/assets/papers/Ultra_Large_Sites_SS09-Seeger_Key_Value_Stores.pdf, Abruf: 14.01.2015

Senatsverwaltung für Wirtschaft, Technologie und Forschung (o.J.): Datensätze,
http://daten.berlin.de/datensaetze, Abruf: 01.01.2015

Senatsverwaltung für Wirtschaft, Technologie und Forschung (2014): Berliner Weihnachtsmärkte 2014,
http://daten.berlin.de/datensaetze/berliner-weihnachtsm%C3%A4rkte-2014,
Abruf: 01.01.2015

Stadt Köln (o.J.): Datasets,
http://www.offenedaten-koeln.de/dataset, Abruf: 01.01.2015

Sunlight Foundation (2010): Ten principles for opening up government information,
http://sunlightfoundation.com/policy/documents/ten-open-data-principles/,
Abruf: 04.01.2014

Vogels, W. (2012): Amazon DynamoDB – a Fast and Scalable NoSQL Database Service Designed for Internet Scale Applications,
http://www.allthingsdistributed.com/2012/01/amazon-dynamodb.html,
Abruf: 14.01.2015

Wer denkt was GmbH (2013): Liste von deutschen Strassennamen,
http://www.datendieter.de/item/Liste_von_deutschen_Strassennamen_.csv,
Abruf: 06.01.2015

Wikipedia (2014a): PuTTY,
http://de.wikipedia.org/wiki/PuTTY,
Abruf: 21.12.2014

Wikipedia (2014b): WinSCP,
http://de.wikipedia.org/wiki/WinSCP,
Abruf: 09.01.2015

Wikipedia (2014c): Node.js,
http://de.wikipedia.org/wiki/Node.js,
Abruf: 25.12.2014

World Bank Group (o.J.): Data Catalog,
http://datacatalog.worldbank.org/, Abruf: 28.12.2014

World Bank Group (2014): World Development Indicators,
http://data.worldbank.org/data-catalog/world-development-indicators,
Abruf: 20.12.2014



Konzepte und Einsatzszenarien von Wide-Column-Datenbanken

Schriftliche Ausarbeitung
im Rahmen der Lehrveranstaltung „Integrationsseminar“
für das Kompetenzzentrum Open Source (KOS)

Vorgelegt von

Maximilian Weißbach, Tobias Knorr,
Marcel Ruoff

am 23.01.2015

Fakultät Wirtschaft
Studiengang Wirtschaftsinformatik
WWI2012V

Inhaltsverzeichnis

Abkürzungsverzeichnis

API	Application Programming Interface
CMD	Kommandozeilenprogramm
CQL	Continuous Query Language
CSV	Comma-Separated Values
DBMS	Datenbankmanagementsystem
IDC	International Data Corporation
JDK	Java Development Kit
JRE	Java Runtime Environment
MVCC	Multiversion Concurrency Control
NSA	National Security Agency
OLAP	Online Analytical Processing
RDBMS	Relational Database Management System
REST	Representational State Transfer

Abbildungsverzeichnis

1 Einführung

Große Firmen, wie Google und Facebook, wurden in den letzten Jahren mit immer größeren Datenmengen konfrontiert. Das weltweite Netz und seine Daten vergrößern sich explosionsartig. Nach derzeitigen Schätzungen wird der Datenbestand eine Größe von etwa 35 Zetabyte bis zum Jahr 2020 erreicht haben.[1] Der Umstand, dass die Daten immer mannigfacher werden, erfordert flexiblere Herangehensweisen. E-Mails, Dokumente, Streams können nicht mehr in den klassischen Datenbanksystemen verwaltet werden.[2] Um den Anforderungen zu begegnen, entwickelten die oben genannten Firmen zusammen mit der freien Community und Open-Source-Entwicklern NoSQL-Datenbanken. Ziel war es neue Methoden für Speicherung, Suche, Schutz, Sicherheit und Überwachung zu finden. [3]

NoSQL stellt das relationale Modell als absolute Lösung für alle Probleme in Frage. Als NotOnlySQL steht es dennoch nicht für eine völlige Ablehnung von SQL. [4]

Eine andere Möglichkeit das Problem der steigenden Datenmengen zu lösen, die von relationalen Datenbanken angeboten wird, ist die vertikale Skalierung: Mehr Prozessoren, mehr Speicher und größere Festplatten. Dieses Vorgehen ist jedoch beschränkt. Im Gegensatz dazu setzen NoSQL-Datenbanken auf eine horizontale Skalierbarkeit.[5]

Im Laufe der Zeit wurde versucht, immer größere Datenmengen immer besser zu analysieren. Staatliche Akteure, Wirtschaftsprüfer, Controlling, Kunden und Endnutzer profitierten von der immer effektiveren Informationsverarbeitung. Durch die Entwicklung des DataMining und ähnlicher Disziplinen musste weiter versucht werden, einen hohen Mehrwert aus gesammelten Daten zu erzielen. Viele Unternehmen basieren allein auf der Geschäftsgrundlage, dass sie ihre Nutzer und deren Verhalten möglichst genau kennen und legen somit Wert auf effiziente Datenbank-Typen.[6]

Ende der 80er-Jahre wurde die Idee der Wide-Column-Stores entwickelt. Diese werden in der Praxis für analytische Informationssysteme wie DataMining, Business Reporting, Prozess-Management und Reporting eingesetzt.[7]

[1] Vgl. FH Köln (2014).
[2] Vgl. Walker-Morgan (2010).
[3] Vgl. FH Köln (2014).
[4] Vgl. Walker-Morgan (2010).
[5] Vgl. FH Köln (2014).
[6] Vgl. FH Köln (2014).
[7] Vgl. FH Köln (2014).

Diese Ausarbeitung befasst sich einerseits mit einer Vorstellung der theoretischen Grundlage und Konzepte der Wide-Column-Datenbanken und ordnet diese in die NoSQL Welt ein. Andererseits sollen die möglichen Einsatzszenarien ermittelt und vorgestellt werden.

Deshalb beginnt diese Arbeit mit der Vorstellung der vier Hauptgruppen von NoSQL-Systemen. Anschließend erfolgt die Beschreibung der theoretischen Grundlagen, um ein Verständnis für Wide-Column-Datenbanken zu schaffen. Nachdem diese Grundlage geschaffen wurde, erfolgt eine Hinführung zu den Wide-Column-Datenbanken. Es wird die allgemeine Struktur der Wide-Column-Datenbank vorgestellt und im Anschluss daran drei ausgewählte Wide-Column-Datenbanksysteme betrachtet. Diese werden abschließend in einem Kriterienkatalog gegenübergestellt. Die Implementierung eines Datenbankprototyps und letztlich der Vergleich mit einer relationalen Datenbank bilden den Praxisteil. Im Fazit werden abschließend die gewonnenen Erkenntnisse dieser Arbeit reflektiert.

2 Grundlagen von NoSQL-Datenbanken

Dieses Kapitel geht auf die Begriffe Big Data und NoSQL ein und grenzt diese zugleich voneinander ab.

2.1 Big Data und Grenzen von RDBMS

Laut einer Studie der International Data Corporation (IDC) waren im Jahr 2012 2,7 Zettabytes digitale Daten weltweit vorhanden. Diese großen Datenmengen, die mit Hilfe von Relational Database Management Systems (im weiteren Verlauf RDBMS genannt) nicht mehr oder nur noch ungenügend verarbeitet, gespeichert, geschützt und effizient für Analysen und strategische Szenarien zur Verfügung gestellt werden können, nennt man Big Data.[8]

Dieser Begriff definiert sich dabei durch das sog. 4-V-Modell. Es beschreibt die Herausforderungen des Datenwachstums. Die vier Dimensionen sind:[9]

- Volume: Steigendes Volumen der Daten
- Velocity: Steigende Geschwindigkeit der Erzeugung von Daten
- Variety: Steigende Vielfalt der erzeugten Daten
- Veracity: Zuverlässigkeit von Daten[10]

Das steigende Volumen, entsteht mitunter durch die Unternehmensprozesse (Bsp. Logfiles, Transaktionsdaten oder Produktionsdaten)[11] oder auch durch die Datensammlung von Institutionen.[12] Darüber hinaus sorgt das Web 2.0 für einen enormen Anstieg der Daten. Ein Vertreter dieser neuen Web Generation ist Facebook und dessen Nutzer erzeugen pro Minute ca. 650.000 verschiedene Inhalte oder verteilen ca. 35.000 „Likes“. Diese Datenmenge stellt relationale Datenbanken vor große Herausforderungen. Es gibt zwar relationale Datenbanken die bis in den Petabyte Bereich arbeiten können, doch diese sind sehr teuer.[13] Relationale Datenbanken können nur vertikal skaliert werden, um die Performance zu erhöhen. Dies bedingt aber eine technische Grenze und darüber hinaus nimmt der Preis überproportional zur Leistung zu.[14] Relationale Datenbanken sind durch ihre Architektur für häufige Transaktionen auf Datensatzebene oder für den Einsatzbereich mit kleinen bis mittleren Da-

---

[8] Vgl. Klein, D. / Tran-Gia, P. / Hartmann, M. (2013), vgl. auch dazu Pientka, F. (2011), vgl. auch dazu Niemann, C. (2011)
[9] Vgl. Klein, D. / Tran-Gia, P. / Hartmann, M. (2013)
[10] Vgl. ebenda
[11] Vgl. Niemann, C. (2011)
[12] Vgl. ebenda
[13] Vgl. Klein, D. / Tran-Gia, P. / Hartmann, M. (2013)
[14] Vgl. Wolff, E. (2012)

tenvolumen vorgesehen.[15] Deshalb nimmt die Performance von RDBMS bei wachsender Datenmenge stetig ab.

Die steigende Geschwindigkeit betrachtet dabei zwei unterschiedliche Aspekte. Der erste Aspekt bezieht sich auf die Rate, mit der neue Daten erstellt werden. Diese nimmt in den verschiedensten Bereichen zu. Der zweite Aspekt betrachtet die Weiterverarbeitung der Daten, um schnellstmöglich auf diese reagieren zu können. Ein Beispiel stellt die Onlineplattform YouTube dar. Pro Minute werden ca. 30 Stunden Videomaterial hochgeladen und zeitgleich ca. 1,3 Millionen Videos konsumiert.[16]

Der wohl wichtigste Punkt in der Definition von Big Data ist die Vielfalt der erzeugten Daten. Die unterschiedlichen und nicht strukturierten Daten stellen RDBMS, die nur effizient mit strukturierten Datensätzen arbeiten können, vor große Herausforderungen. Ein Beispiel stellen E-Mails dar. Sie sind halbstrukturiert, da Empfänger und Betreff strukturiert sind, doch deren Inhalt kann unterschiedliches enthalten. Bei Big Data jedoch, werden sowohl strukturierte, halbstrukturierte als auch nicht strukturierte Datensätze in Datenbanken gespeichert und ausgewertet. Diese Datenansammlung kann dabei nach drei Kategorien klassifiziert werden.[17]

- zwischen Personen (z. B. soziale Netzwerke)
- zwischen Personen und Diensten/Maschinen (z. B. E-Commerce Anwendungen)
- zwischen Diensten/Maschinen (z. B.. GPS Positionsinformationen)[18]

Der letzte Definitionspunkt bezieht sich auf die Zuverlässigkeit der Daten. Da Analysen häufig schnellstmöglich zur Verfügung stehen müssen, können die Daten oftmals nicht bereinigt werden und sind damit ungenau. Diese Ungenauigkeit muss ebenfalls berücksichtigt werden.[19]

2.2 Definition von NoSQL-Datenbanken

Der homogene Begriff NoSQL ist bis heute noch nicht genau definiert, da es weder Gremien noch Organisationen gibt, die sich mit der Klärung des Begriffes auseinandergesetzt haben. Deshalb versteht bis heute jeder etwas anderes unter diesem Begriff.[20]

Der Begriff NoSQL ist irreführend, da man annehmen könnte, dass nicht relationale Datenbanken auf SQL gänzlich verzichten. Dies ist nicht der Fall, denn es gibt verschiedene

[15] Vgl. Niemann, C (2011)
[16] Vgl. Niemann, C (2011)
[17] Vgl. Klein, D. / Tran-Gia, P. / Hartmann, M. (2013)
[18] Vgl. ebenda
[19] Vgl. ebenda
[20] Vgl. Edlich, S. u. a. (2010), S. 2

NoSQL-Datenbanken, wie OrientDB, die diese Sprache implementiert haben.[21] NoSQL wird eher als Synonym für relationale Datenbanken verstanden und müsste deshalb eher „No-REL“ heißen, wie Carlo Strozzi es vorschlug. Doch der Begriff NoSQL entwickelte sich schnell zum Markennamen und war nicht mehr aus der Fachwelt zu verdrängen. Die NoSQL-Community erweiterte den Begriff NoSQL um Not only SQL und gab ihm so eine zutreffendere Bezeichnung. Durch diese Bezeichnung wird klar, dass NoSQL-Datenbanken mit den bisherigen Eigenschaften von RDBMS brechen.[22]

Im NoSQL-Archiv wird anhand einer Definition versucht, NoSQL-Datenbanken von relationalen Datenbanken zu unterscheiden. Demnach versteht man unter NoSQL-Datenbanken eine neue Generation von Datenbanken, die folgende Merkmale aufweisen:[23]

1. Datenmodell ist nicht relational
2. Verteilte und horizontale Skalierbarkeit
3. Open Source
4. Schemafrei
5. Einfache Datenreplikation
6. Einfache API
7. Eventually Consistent / BASE

Da es verschiedene NoSQL-Datenbanken gibt, deckt auch nicht jede NoSQL-Datenbank alle Merkmale gleichermaßen ab.[24]

2.3 Kategorisierung von NoSQL-Systemen

Aktuell befinden sich 150 verschiedene NoSQL-Datenbanken im NoSQL-Archiv.[25] Da diese Datenbanken in ihrem System nicht homogen sind, wird im Folgenden eine Untergliederung der verschiedenen NoSQL-Datenbanksysteme vorgenommen.

Eine Grobgliederung bilden die sog. Core-NoSQL-Systeme und die Other-NoSQL-Systeme in Abb. 1.

[21] Vgl. Thurm B. (2012), S. 3
[22] Vgl. Edlich, S. u. a. (2010), S. 4 f., vgl. auch dazu Thurm B. (2012), S. 3 f.
[23] Vgl. NoSQL-Database (o. J.)
[24] Vgl. Edlich, S. u. a. (2010), S. 2 ff.
[25] Vgl. NoSQL-Database (o. J.)

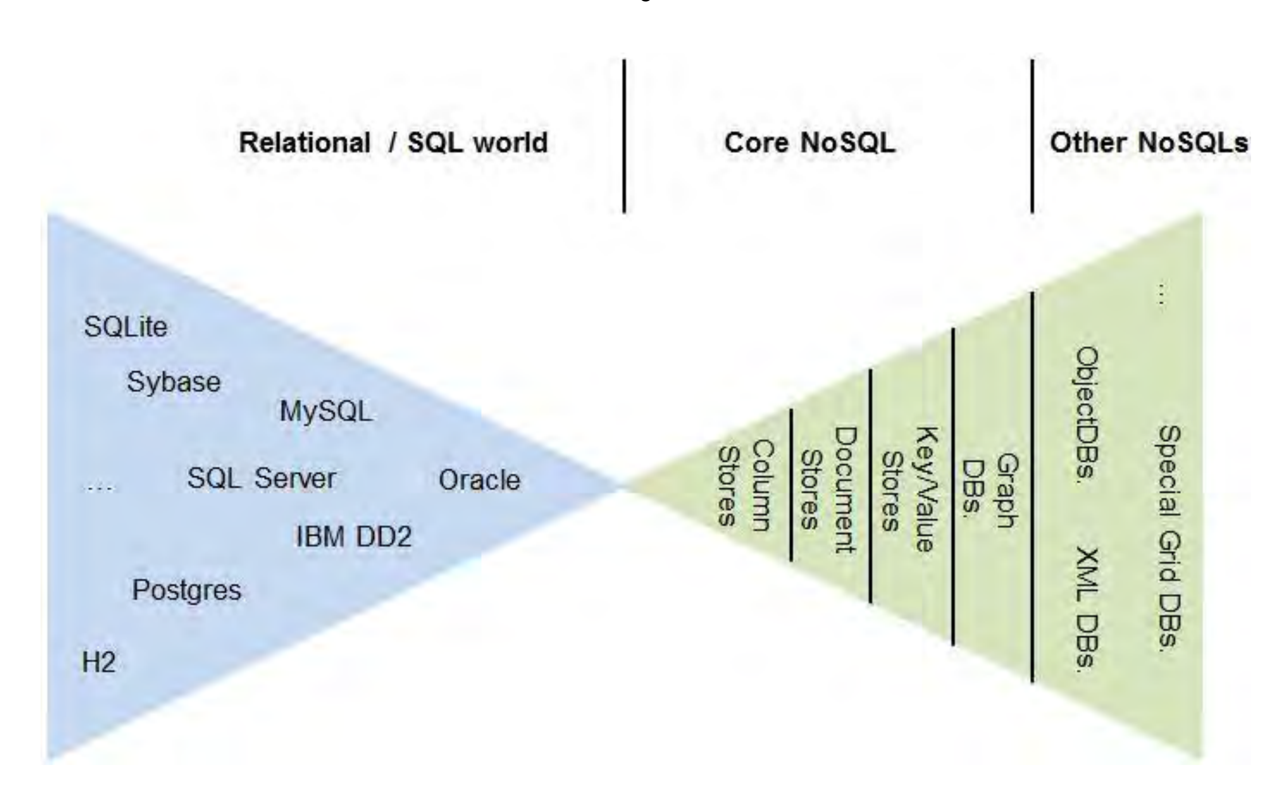


Abb. 1: SQL- und NoSQL-Systeme[26]

Da es den Rahmen dieses Projektes sprengen würde, wird lediglich auf die Core-NoSQL-Systeme eingegangen.

2.3.1 Key/Value-Stores

Dem Key/Value-Systeme liegt ein einfaches Schlüssel- und Wertschema zugrunde. Einem Wert (Value) ist dabei immer ein bestimmter Schlüssel (Key) zugeordnet. Die Schlüssel können in Namensräumen und Datenbanken aufgeteilt werden. Die Werte können zusätzlich zu Strings auch Listen, Sets oder Hashes enthalten. Darüber hinaus können die Key/Value-Systeme in zwei Gruppen untergliedert werden. Die In-Memory-Variante sorgt für eine hohe Performanz, da sie die Daten im Arbeitsspeicher behält. Dadurch bieten sich solche Datenbanken als verteilte Cache-Speichersysteme an. Vertreter dieser Gruppe sind: Redis, Oracle Coherence.

Die On-Disk-Variante speichert die Daten auf der Festplatte. Diese Datenbanken bieten sich somit als Datenspeicher an. Vertreter dieser Gruppe sind: BigTable, Redis, MongoDB.[27]

[26] Mit Änderungen entnommen aus: Edlich, S. u. a. (2010), S. 6
[27] Vgl. Edlich, S. u. a. (2010), S. 7 , vgl. auch dazu Ngoc Ha, T. (2011), S. 4

2.3.2 Wide-Column-Store / Column-Family-Systeme

Dieses System ähnelt einer Excel-Tabelle. Dabei werden die Daten nach ihren Spalten und nicht nach ihren Zeilen gruppiert. Hier können beliebig viele Schlüssel auf beliebig viele Key/Value Paare angewendet werden. Eine Spalte kann darüber hinaus mit beliebig vielen Key/Value-Paaren erweitert werden. Diese Key- und Value-Listen können dann meistens noch in Form von Keyspaces oder Clustern organisiert werden. Ein Eintrag besteht aus dem Namen, der Spalte, die Daten und einem Zeitstempel zur Aktualitätsprüfung. Dabei bilden die zusammenhängenden Spalten die Column-Family. Durch die Schemafreiheit, kann ein Datensatz Spalten mit zuvor nicht definierten Namen und Typ besitzen. Darüber hinaus kann die Anzahl von Spalten bei jedem Datensatz unterschiedlich sein. Dieses System bildet somit eine Mischung aus Key/Value-System und spaltenorientierter Datenbank oder auch einer relationalen Datenbank.[28]

Die wichtigsten Vertreter dieses Datenbanktyps sind: Cassandra, HBase, Accumulo, Hypertable und Sqrrl.[29]

2.3.3 Document Stores

Die Document-Stores zeichnen sich durch eine schemafreie Organisation der Daten aus. Die Daten werden in Form von Dokumenten gespeichert. Ein Dokument entspricht dabei nicht einer z. B. .txt Datei, sondern man versteht eine Zusammenstellung von strukturierten Daten. Diese können ein Datensatz (Tupel), Tabelle oder auch Objekte enthalten. Durch die schemafreie Organisation der Datenbank, kann jedes Dokument unterschiedlich strukturiert sein. D. h. jedes Dokument ist eine geschlossene Einheit, die ein beliebiges Schema und einen beliebigen Inhalt aufweist. Die Daten in einem solchen Dokument werden über Key-Value-Paare gespeichert. Dabei kann der Wert (Value) einen beliebigen Typ entsprechen z. B. String oder Array. Die Dokumente sind in der Datenbank über Bezeichner gekennzeichnet, über die gleichzeitig zugegriffen werden kann. Darüber hinaus gibt es zwischen den einzelnen Dokumenten keine Relationen.[30]

Die wichtigsten Vertreter dieses Datenbanktyps sind: MongoDB, CouchDB, Couchbase, MarkLogic und RavenDB.[31]

---

[28] Vgl. Edlich, S. u. a. (2010), S. 7 vgl. auch dazu DB-Engines (o. J. b), vgl. auch dazu Ngoc Ha, T. (2011), S. 7 f.
[29] Vgl. DB-Engines (2014c)
[30] Vgl. Edlich, S. u. a. (2010), S. 8, vgl. auch dazu Ngoc Ha, T. (2011), S. 5 f.
[31] Vgl. DB-Engines (2014a)

2.3.4 Graphdatenbanken

Es gibt eine Vielzahle verschiedener Arten von Graphdatenbanken, sowie es verschiedene Arten von Graphen gibt. Dabei bestehen die Graphendatenbanken aus Knoten und Kanten. Die Knoten entsprächen dabei einem Tupel aus dem RDBMS und die Kanten sind die Beziehungen zwischen den Knoten. Die Datenbanken nutzen Traversierungen (durchlaufen von Knoten und Kanten) um performante Abfragen zu gewährleisten.[32]

Die wichtigsten Vertreter dieses Datenbanktyps sind: Neo4j, Titan, OrientDB, Sparksee und Giraph.[33]

[32] Vgl. Edlich, S. u. a. (2010), S. 8 f., vgl. auch dazu Ngoc Ha, T. (2011), S. 9 f.
[33] Vgl. DB-Engines (2014b)

3 Theoretische Grundlagen für NoSQL

3.1 Map/Reduce

Durch das Map/Reduce-Framework wird eine nebenläufige Berechnung extrem großer Datenmengen ermöglicht. Der Map/Reduce-Algorithmus ist an die funktionale Programmierung angelehnt, in der mit map() und reduce() Routinen gearbeitet wird. Dabei wird der Algorithmus auf Listen angewendet. Dieser Algorithmus lässt sich dabei in zwei Phasen einteilen, die zu den Funktionen höherer Ordnung gehören.[34]

Die Funktion map() arbeitet alle Listenelemente mit einer vorgegebenen Funktion ab und gibt eine modifizierte Liste zurück. Die Funktion reduce() trägt die Funktionsergebnisse der Listenpaare zusammen und reduziert diese auf einen Ausgabewert. Da der Ursprung in der funktionalen Programmierung liegt, lässt sich eine Parallelisierung in den Map/Reduce-Phasen vornehmen. Der Anwender muss die Map-Funktion und die Reduce-Funktion, sprich die Logik seiner Anwendung, selber erstellen. Wohingegen das Map/Reduce-Framework die Verteilung der Daten, von Map- und Reduce-Prozessen sowie die Speicherung der Zwischen- und Endergebnisse vornimmt. Darüber hinaus ermöglicht das Framework Parallelisierung, Fehlertoleranz, Datentransfer, Lastverteilung und Monitoring.[35]

Bei dem Map/Redurce Verfahren lassen sich folgende Arbeitsphasen unterscheiden (siehe Abb. 2).

- „Zuerst werden alle Eingabedaten auf verschiedene Map-Prozesse aufgeteilt.
- In der Map-Phase berechnen diese Prozesse jeweils parallel die vom Nutzer bereitgestellte Map-Funktion.
- Von jedem Map-Prozess fließen Daten in verschiedene Zwischenergebnisspeicher.
- Die Map-Phase ist beendet, sobald alle Zwischenergebnisse berechnet worden sind.
- Nun beginnt die Reduce-Phase, in der für jeden Satz an Zwischenergebnissen ein Reduce-Prozess die vom Nutzer bereitgestellte Reduce-Funktion parallel berechnet.
- Jeder dieser Reduce-Prozesse speichert seine Ergebnisdatei im Dateisystem ab.
- Die Reduce-Phase ist beendet, sobald alle Ergebnisdateien abgespeichert worden sind. Somit ist auch die gesamte Durchführung des Map/Reduce-Verfahrens beendet.“[36]

[34] Vgl. Edlich, S. u. a. (2010), S. 12 f.
[35] Vgl. ebenda, S. 13 ff.
[36] Ebenda, S. 17

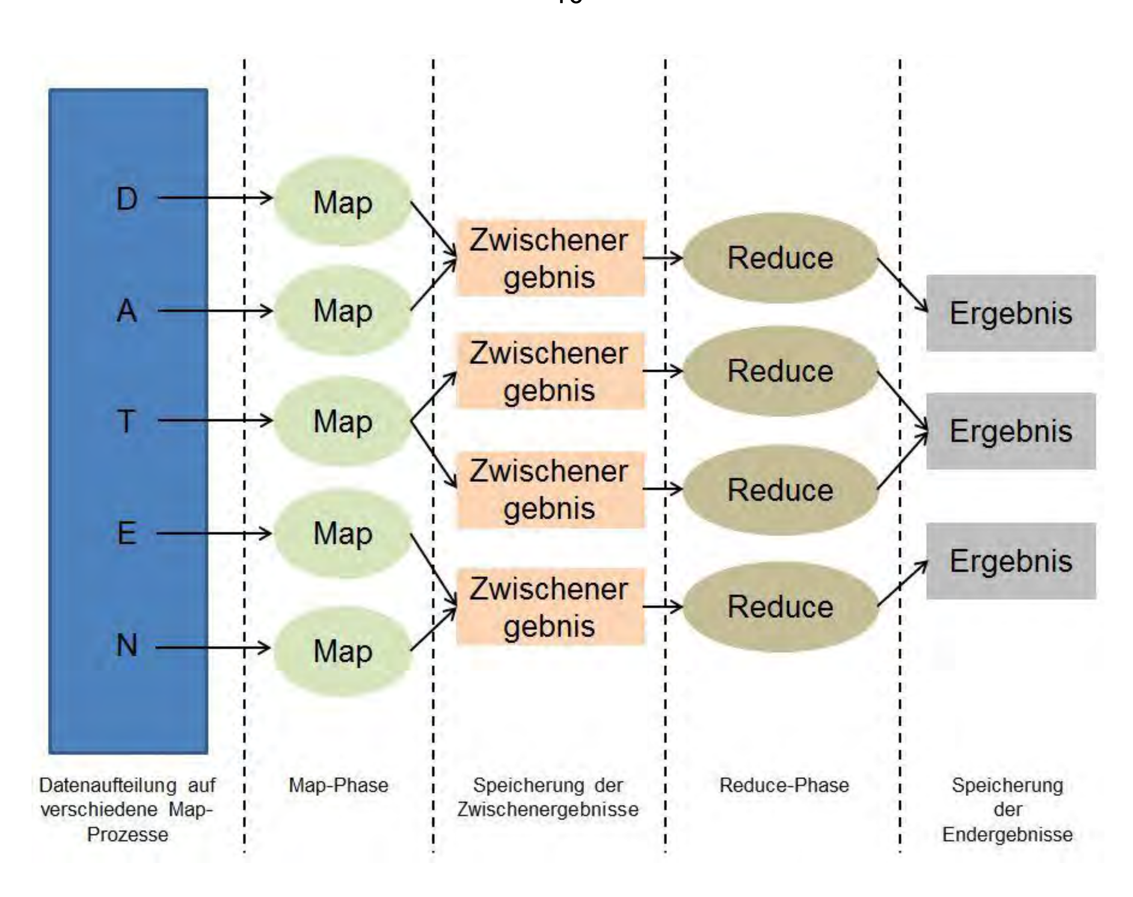


Abb. 2: Map/Reduce-Verfahren[37]

3.2 CAP-Theorem und das BASE-Modell

Um das CAP-Theorem zu verstehen, muss man zuvor die Bedeutung der Abkürzung CAP verstehen. So steht C für Consistency (Konsistenz), A für Availability (Verfügbarkeit) und P für Partition Tolerance (Ausfalltoleranz).

Konsistenz bedeutet, dass nach Abschluss einer Transaktion ein konsistenter Zustand in der verteilten Datenbank vorliegt. In einer verteilten Datenbank mit replizierenden Knoten ist dies erst dann der Fall, wenn alle Knoten den neuen Wert zurückliefern.

Verfügbarkeit bedeutet, dass eine akzeptable Reaktionszeit vorliegt. Diese kann von Anwendung zu Anwendung variieren.

Ausfalltoleranz bedeutet, dass das gesamte System immer noch auf Anfragen reagiert, obwohl ein Knoten oder eine Kommunikationsverbindung ausgefallen ist.

Nach Brewers Kernaussage des CAP-Theorems können maximal zwei dieser drei genannten Kriterien erreicht werden - Abb. 3.[38]

[37] Mit Änderungen entnommen aus: Edlich, S. u. a. (2010), S. 18
[38] Vgl. Edlich, S. u. a. (2010), S. 31 f.

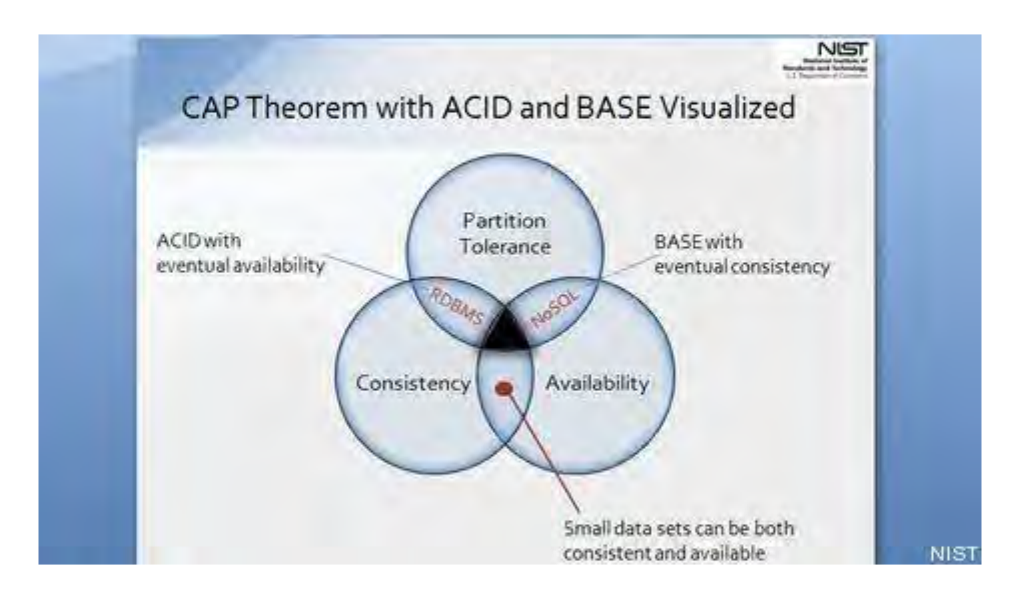


Abb. 3: CAP-Theorem[39]

Um diesen Konflikt des CAP-Theorems zu lösen, bedarf es eines neuen Modelles, welches die Konsistenz in verteilten Datenbanksystemen betrachtet. Dieses BASE-Modell (Basically Available, Soft State, Eventually Consistent) ist der Gegenpart zum ACID-Modell (Atomicity, Consistency, Integrity, Durability). Wie man schon an den Begrifflichkeiten erkennen kann, ist bei dem BASE-Modell die Verfügbarkeit im Vordergrund, während bei dem ACID-Modell die Konsistenz an erster Stelle steht. Dies ist der Kernunterschied dieser beiden Modelle. Bei dem BASE-Modell wird nach einer gewissen Zeit der Inkonsistenz der Zustand der Konsistenz erreicht. Dies hängt mitunter von der Anzahl der Knoten im System, den durchschnittlichen Reaktionszeiten und der durchschnittlichen Auslastung des Systems ab.[40]

Jede Datenbank kommt dabei dem ACID-Modell oder dem BASE-Modell näher.[41]

3.3 Consistent-Hashing

Consistent-Hashing gehört zu den Streuwertfunktionen und definiert, nach welcher allgemeinen Vorschrift ein Wert x aus einer großen Quellenmenge auf einen Hashwert v = h(x) aus einer kleineren Wertemenge abgebildet wird. Die Anwendungsmöglichkeiten von Hashfunktionen sind sehr vielfältig und deshalb wird nur auf die Berechnung eines Speicherortes für ein Objekt eingegangen.[42]

Da bei verteilten Systemen häufig mehrere Knoten zum Einsatz kommen, ist es eine wichtige Aufgabe des Consistent-Hashing, dass alle Objekte auf n Server verteilt werden.

[39] Enthalten in: MacFadden, G. (2013)
[40] Vgl. Edlich, S. u. a. (2010), S. 33 f.
[41] Vgl. ebenda, S. 34
[42] Vgl. ebenda, S. 36 f.

Darüber hinaus berücksichtigt Consistent-Hashing verschiedene Anforderungen. Zum einen soll bei dem Hinzufügen eines Servers in einem verteilten System, nur ein geringer Teil der im Gesamtsystem gespeicherten Objekte umkopiert werden. Zum anderen soll beim Entfernen eines Servers nur die bislang dort enthaltenen Objekte auf die restlichen Server verteilt werden. Außerdem trägt das Consistent-Hashing der individuellen Speicherkapazität der einzelnen Server Rechnung, anstatt alle Objekte gleichmäßig auf n Server zu verteilen.[43]

Zum besseren Verständnis wird das Hashing-Verfahren anhand einer kreisförmigen Darstellung erläutert - Abb. 4.

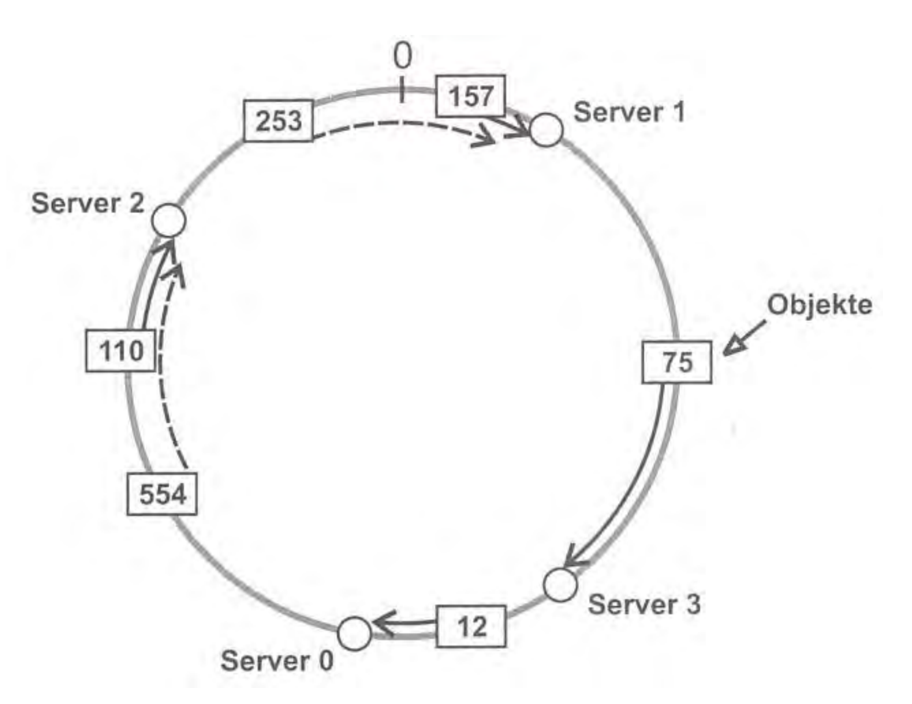


Abb. 4: Abbildung von Servern und Objekten auf einen Ring[44]

Es wird ein Adressraum definiert, in dem später alle Objekte vom Typ TValue anhand des Schlüssels k vom Type TKey gespeichert werden. Danach werden alle Server anhand des berechneten Hashwertes in den Adressraum eingefügt. Zuletzt wird den Objekten ebenfalls anhand ihres Hashwertes ein Platz im Adressraum zugewiesen. Dabei ergibt sich der Speicherort (Server) für ein Objekt aus dem im Uhrzeigersinn nächstgelegenen Hashwert eines Servers. Wird ein Server hinzugefügt, so übernimmt dieser alle Objekte zwischen dem Hashwert seines Vorgängers und dem seines eigenen. Wird hingegen ein Server entfernt, so kopiert dieser alle bei ihm gespeicherten Objekte zu seinem Nachfolger. Um der jeweiligen Leistungsfähigkeit eines Servers gerecht zu werden, wird je nach dessen Leistungsfähigkeit eine größere Anzahl an Hashwerten für ihn berechnet und im Adressraum eingetragen.[45]

[43] Vgl. Edlich, S. u. a. (2010), S. 36 f.
[44] Enthalten in: Edlich, S. u. a. (2010), S. 38
[45] Vgl. Edlich, S. u. a. (2010), S. 37 ff.

3.4 Multiversion Concurrency Control

Durch das Multiversion Concurrency Control Verfahren (MVCC), wird die Integrität der Datensätze sichergestellt. Konkurrierende Zugriffe auf die Datensätze, werden durch eindeutige Versionierungen der Datensätze bei jedem Schreibzugriff kontrolliert. Jeder Schreibzugriff bekommt dementsprechend seine eigene Versionsnummer. Dabei verweist jede Version auf ihre Vorgängerversion, wodurch eine zeitliche Abfolge entsteht.[46] Im Gegensatz zum relationalen Transaktionsansatz, welcher auf Sperren von Datensätzen beruht, können beim MVCC Verfahren weiterhin Lesezugriffe ausgeführt werden. Außerdem führt dieser Ansatz zu erheblichen Performanceverbesserungen, da das Setzen und Aufheben von Sperren entfällt. Somit eignet es sich hervorragend für Datensätze jenseits des Tera-Byte Bereichs.[47]

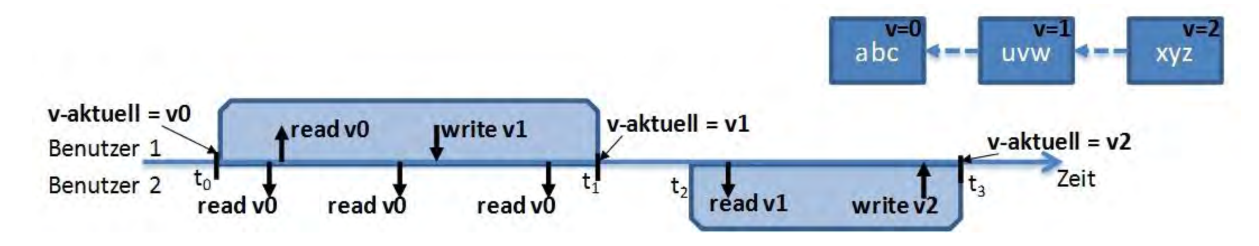


Abb. 5: Lesezugriff MVCC[48]

Wie in Abb. 5 zu sehen ist, steht zu jedem Zeitpunkt eine aktuelle Version zum Lesen bereit. Der Lesezugriff ist komplett vom Schreibzugriff unabhängig.[49]

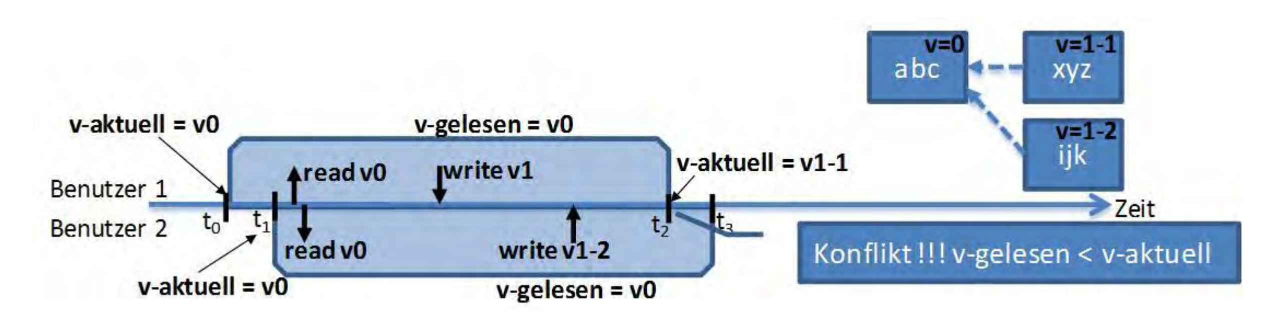


Abb. 6: Schreibzugriff MVCC mit Versionskonflikt[50]

In Abb. 6 wird eine Konfliktsituation zweier Schreibzugriffe dargestellt. Das System überprüft die Vorgänger-Versionsnummer des aktuell in dieser Transaktion geänderten Datensatzes mit der aktuellen Versionsnummer. Stimmen diese überein, wird die seine Versionsnummer zur neuen aktuellen Versionsnummer. Ist die Vorgängerversionsnummer jedoch kleiner als

[46] Vgl. ebenda, S.44 ff.
[47] Vgl. FH Köln (2011)
[48] Enthalten in: FH Köln (2011)
[49] Vgl. ebenda
[50] Enthalten in: FH Köln (2011)

die aktuelle Versionsnummer, kommt es zu einem Konflikt. In diesem Fall können zwei unterschiedliche Situationen auftreten:[51]

1. Das DBMS ist in der Lage die Attribute der neuen Versionen zu vergleichen. Wurden unterschiedliche Attributwerte geändert, können diese Attributwerte beider Transaktionen automatisch zusammengesetzt werden.
2. Wurde das gleiche Attribut geändert, entsteht ein unlösbarer Konflikt, wodurch die Transaktion abgebrochen und eine Mitteilung an den Anwender geschickt wird.

Durch den hohen Speicherbedarf der Versionierung, müssen in regelmäßigen Abständen alte Versionen gelöscht werden, um Speicherkapazitäten freizugeben. [52]

3.5 Representational State Transfer

Representational State Transfer (REST) steht für einen modernen (Web-)Architekturstil. Dieser wurde von Roy Fielding entwickelt. Ziel war es, ein einheitliches Konzept sowohl für statische als auch dynamisch berechnete Informationen zu erstellen. Er definierte 5 Kernprinzipien:[53]

1. Ressourcen mit eindeutiger Identifikation
2. Verknüpfungen/Hypermedia
3. Standardmethoden
4. Unterschiedliche Repräsentation
5. Statuslose Kommunikation

Bei jedem Kommunikationsvorgang werden alle für die Anfrage benötigten Informationen (Login-Daten, eventuelle Einschränkungen, etc.) vom Client übertragen. Dadurch ist ein Dienst der auf der REST-Architektur beruht (auch als RESTful bezeichnet) zustandslos. Durch die vollständige Übertragung jeglicher Informationen, muss keine dauerhafte Verbindung zwischen Server und Client herrschen. Jede Anfrage kann dadurch von jedem anderen Server beantwortet werden, wodurch die Skalierbarkeit gefördert wird.[54]

---

[51] Vgl. FH Köln (2011)
[52] Vgl. ebenda
[53] Ebenda
[54] Vgl. ebenda, vgl. dazu ausführlich: Fielding, R. (2000)

4 Wide-Column-Datenbanken

4.1 Struktur

Bei den Wide-Column-Datenbanken handelt es sich um spaltenorientierte, nichtrelationale Datenbanken, deren Idee bereits aus den 80er Jahren stammt.[55] Im Gegensatz zum zeilenorientierten Ansatz, der typischerweise in relationalen Datenbankmanagementsystemen (RDBMS) verwendet wird, werden die Daten in Spalten abgespeichert. Eine Spalte oder auch Column genannt, bildet die Basiseinheit und besteht aus sogenannten Key-Value Paaren.[56] Als einfaches Beispiel könnte man z.B. den Namen als Key und Martin als Value nehmen.

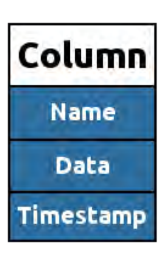


Abb. 7: Spaltenstruktur[57]

Wie in Abb. 7 zu sehen ist, besteht eine Spalte aus einem Namen (Key), den Daten (Value) und einem Zeitstempel. Verwandte oder ähnliche Daten können in einer sogenannten „Column Family“ (zu sehen in Abb. 8) zusammengefasst werden. Eine Column Family ist hierbei gleichbedeutend mit einer Tabelle bei relationalen Datenbanken, folgt allerdings keiner logischen Struktur und kann aus beliebig vielen Spalten bestehen.[58]

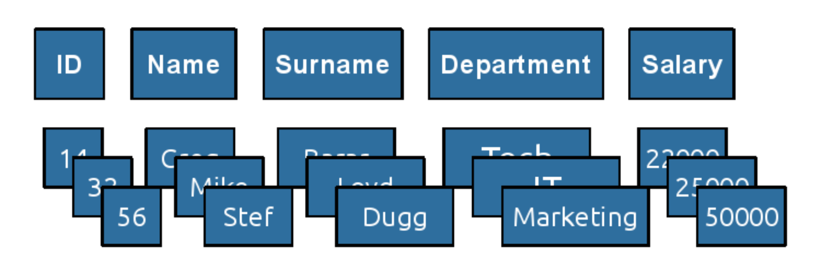


Abb. 8: Darstellung einer Column Family[59]

[55] Vgl. Edelich, S. u.a. (2010), S.53
[56] Vgl. Sadalage, P./Fowler, M. (2013), S.99
[57] Enthalten in: FH Köln (2014)
[58] Vgl. FH Köln (2014)
[59] Enthalten in: FH Köln (2014)

Eine spezielle Form sind die Super Columns. Diese besteht, wie die normale Column auch, aus einem Key Value Paar. Der Unterschied ist, dass die Daten aus einer Liste von Spalten bestehen, welche den Namen als Primärschlüssel verwenden, dies ist in Abb. 9 zu sehen. [60]

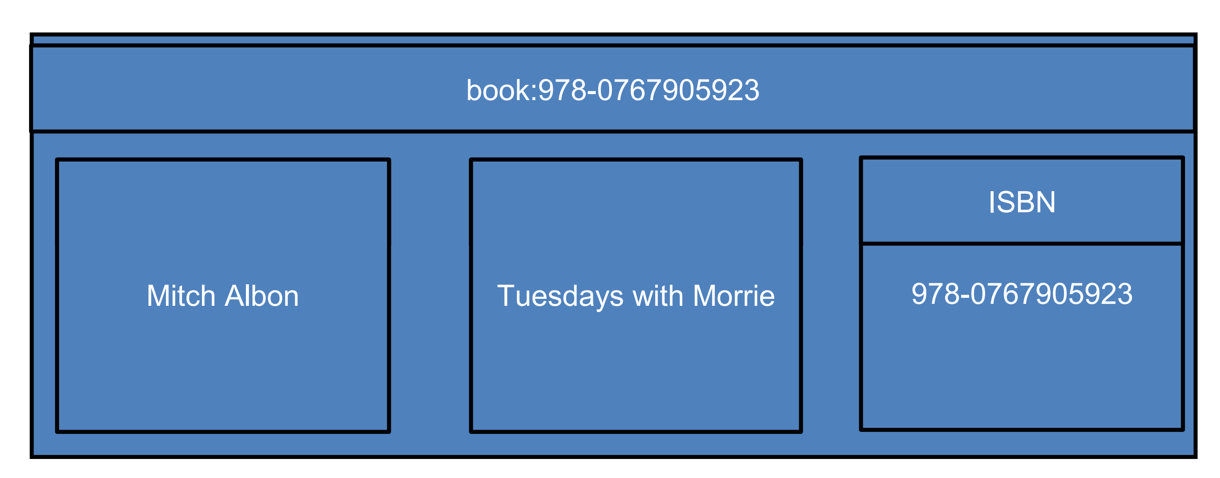


Abb. 9: Super Column[61]

Entsprechend zu der Column Family gibt es auch eine Super Column Family, welche aus einer Ansammlung von Super Columns besteht.

Das Prinzip der Spaltenorientierung eignet sich vor allem zur Aggregation großer Datenmengen, bietet aber auch Vorteile bei der Datenkompression und Analyse der Daten. Diesbezüglich eignet es sich hervorragend für Online Analytical Processing (OLAP) - und Data-Warehouse-Umgebungen.[62] Ein weiteres mögliches Anwendungsgebiet eröffnet sich durch die zunehmende digitale Vernetzung. Eine Wide-Column-Datenbank kann als Event- oder Logliste für einen Kühlschrank eines Autos dienen, in welcher die Rückgabewerte oder Ereignisse von verschiedenen Sensoren gespeichert werden[63]. Dies ist dahingehend sinnvoll, da durch die Schemafreiheit jedwede Datenstruktur in einer Tabelle abgespeichert werden kann. Allerdings führt eine derartige Struktur auch zu einigen Nachteilen. Das Suchen und Einfügen von Daten kann sich als schwieriger erweisen und das Lesen und Schreiben von zusammenhängenden Spaltendaten resultiert in einem aufwendigen springen und suchen, bis alle Daten zur Verfügung stehen.[64]

[60] Vgl. Sadalage, P./Fowler, M. (2013), S. 101
[61] Mit Änderungen entnommen aus: Sadalage, P./Fowler, M. (2013), S. 101
[62] Vgl. Edelich, S. u.a. (2010), S.53
[63] Vgl. Sadalage, P./Fowler, M. (2013), S. 108
[64] Vgl. Edelich, S. u.a. (2010), S.53

4.2 Wide-Column-Datenbankmanagementsysteme

Wie bei relationalen Datenbanken gibt es auch bei den Wide-Column-Datenbanken unterschiedliche Datenbankmanagementsysteme (DBMS). Die Firma solid IT berechnet monatlich den Popularitätswert unterschiedlicher Datenbanken und stellt diesen Online zur Verfügung. Als Berechnungsgrundlage werden nicht die Installationen eines Systems gemessen, sondern es wird auf andere Parameter zurückgegriffen, wie z.B. Anzahl der Jobangebote oder Nennungen des Systems auf Websites gemessen als Treffer in Suchmaschinen, um damit Rückschlüsse auf die Häufigkeit der Installationen zu ziehen.[65]

5 Systeme im Ranking, Dezember 2014

Rang	Vormonat	DBMS	Datenbankmodell	Punkte	Änderung
1.	1.	Cassandra	Wide Column Store	94,06	+2,07
2.	2.	HBase	Wide Column Store	51,08	+4,11
3.	3.	Accumulo	Wide Column Store	2,70	-0,07
4.	4.	Hypertable	Wide Column Store	0,66	+0,02
5.	5.	Sqrrl	Multi-Model	0,19	+0,01

Abb. 10: Rangliste der Popularität von Wide-Column-Datenbankmanagementsystemen[66]

In Abb. 10 sieht man die Popularitätsrangliste von Wide-Column- Datenbankmanagementsystemen. Im Rahmen dieser Ausarbeitung werden die ersten drei Systeme genauer betrachtet, um im Anschluss einen Kriterienkatalog zu erstellen.

4.2.1 Cassandra

	Apache Cassandra
Entwickler	Facebook; Apache Software Foundation (seit 2008)
Letzte Version	Apache Cassandra 2.1.2 (Release 10.11.2014)
Plattform	plattformunabhängig
Lizenz	Open-Source (Version 2 Apache Lizenz)
Einsatz	Twitter
Implementierung	Java

Abb. 11: Steckbrief Apache Cassandra

[65] Vgl. dazu ausführlich: DB-Engines (o. J. a)
[66] Enthalten in: DB-Engines (2014c)

Bei Cassandra handelt es sich um ein ehemals von Facebook entwickeltes Datenbankmanagementsystem, welches 2008 als Open Source freigegeben wurde. Seit Anfang 2010 wurde es von der Apache Software Foundation als Top-Level-Projekt aufgenommen. Wie bei den meisten No-SQL DBMS wurde Googles BigTable als Vorbild genommen.[67] Ziel war es, unter den Aspekten von Performanz und Effizienz die Indizes von Nutzernachrichten zu speichern und zu suchen. Entscheidende Kriterien bei der Entwicklung von Cassandra waren eine hohe Verfügbarkeit und Skalierbarkeit. Die sofortige Konsistenz war dabei weniger entscheidend.[68]

Funktionen

Um die Kriterien einer hohen Verfügbarkeit und Skalierbarkeit zu erfüllen, wurde auf eine verteilte Datenbanklösung gesetzt. Hierbei werden die Daten auf mehrere Rechnerknoten (Cluster) verteilt, wodurch ein „Single Point of Failure" verhindert und eine ständige Erreichbarkeit der Anwendung ermöglicht wird (Zero-Downtime).[69]

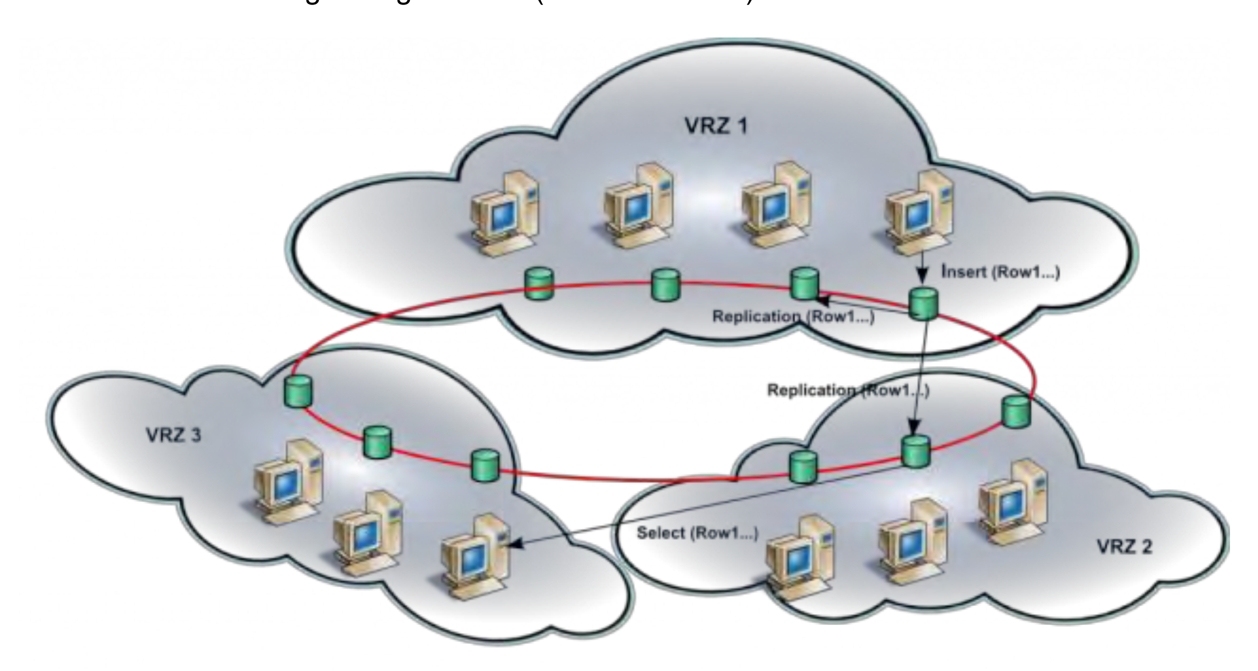


Abb. 12: Casssandra Cluster[70]

In Abb. 12 wird ein Rechnercluster dargestellt. Wie zu sehen ist, werden Schreibvorgänge eines Clients (Inserts) automatisch, nach den definierten Parametern, repliziert (Replication). Beim Zugriff der Daten (Select) wird außerdem ein Loadbalancing eingesetzt. Cassandra arbeitet mit dem Prinzip der Distributed Hash Table. Ein fest definierter Schlüsselraum ver-

[67] Vgl. Edelich, S. u.a. (2010), S.69
[68] Vgl. ebenda, S.70
[69] Vgl. ebenda, S.70
[70] Enthalten in: Hahn Projects (o.J.)

teilt die einzelnen Schlüssel gleichmäßig auf die einzelnen Knoten im Cluster, sodass ein logischer Ring entsteht. Diese Schlüssel dienen dazu, die Daten zu adressieren und über Consistent Hashing Verfahren zur Schlüsselvergabe wird jedem Rechner im Cluster mittgeteilt, welcher Knoten für die Anfrage zuständig ist und kann die Anfrage entsprechend weiterleiten. Wenn ein Knoten aus dem Ring ausfällt, übernimmt ein Replikat seinen Platz bis er wieder zur Verfügung steht.[71]

Vor-Nachteile

Einer der Grundlegenden Vorteile von Cassandra ist das Zero-Downtime Prinzip. Ein weiterer Vorteil sind die hohen Schreib- und Lesearten und die relativ freie Strukturierung der Datensätze. Im Vergleich zu anderen Wide-Column Datenbanken, wie HBase und Accumulo, ist die Installation und Handhabung relativ einfach. Als Nachteil dagegen steht jedoch die beschränkte Auswahl an Abfragemöglichkeiten, welche Cassandra bietet.[72]

4.2.2 HBase

	Apache HBase
Entwickler	Powerset; Apache Software Foundation (seit 2008)
Letzte Version	Apache HBase 2.0.0 (Release 17.12.2014)
Plattform	plattformunabhängig
Lizenz	Open-Source (Version 2 Apache Lizenz)
Einsatz	Facebook
Implementierung	Java

Abb. 13: Steckbrief Apache HBase

HBase ist ein Teilprojekt von Apache Hadoop, einem Projekt der Apache Software Foundation. Nachdem Google 2006 seine Architektur des verteilten Datenbanksystems BigTable veröffentlichte, begann die Entwicklung von HBase. Ziel war es einen quelloffenen Klon von Googles BigTable zu erzeugen. [73]

Funktionen

Wie bei relationalen Datenbanken speichert HBase die Daten in Tabellen mit Zeilen und Spalten, wobei die Zeile für einen Datensatz und eine Spalte für ein Attribut steht. Jede Zeile besitzt einen eindeutigen Schlüssel (hierbei handelt es sich um ein Byte-Array), welcher mit einem Primärschlüssel verglichen werden kann. Die Sortierung der Zeilen erfolgt anhand des

[71] Vgl. Edelich, S. u.a. (2010), S.80 f.
[72] Vgl. ebenda, S.82 f.
[73] Vgl. ebenda, S.55

Schlüssels. Im Gegensatz zu relationalen Datenbanken gibt es bei den Spalten keinen fest definierten Datentyp mit einer bestimmten Länge. Die Zelleninhalte bestehen aus Byte-Arrays, welche auf der Ebene der Applikationslogik in den richtigen Datentyp umgewandelt werden müssen. Leere Zellen existieren nicht.[74]

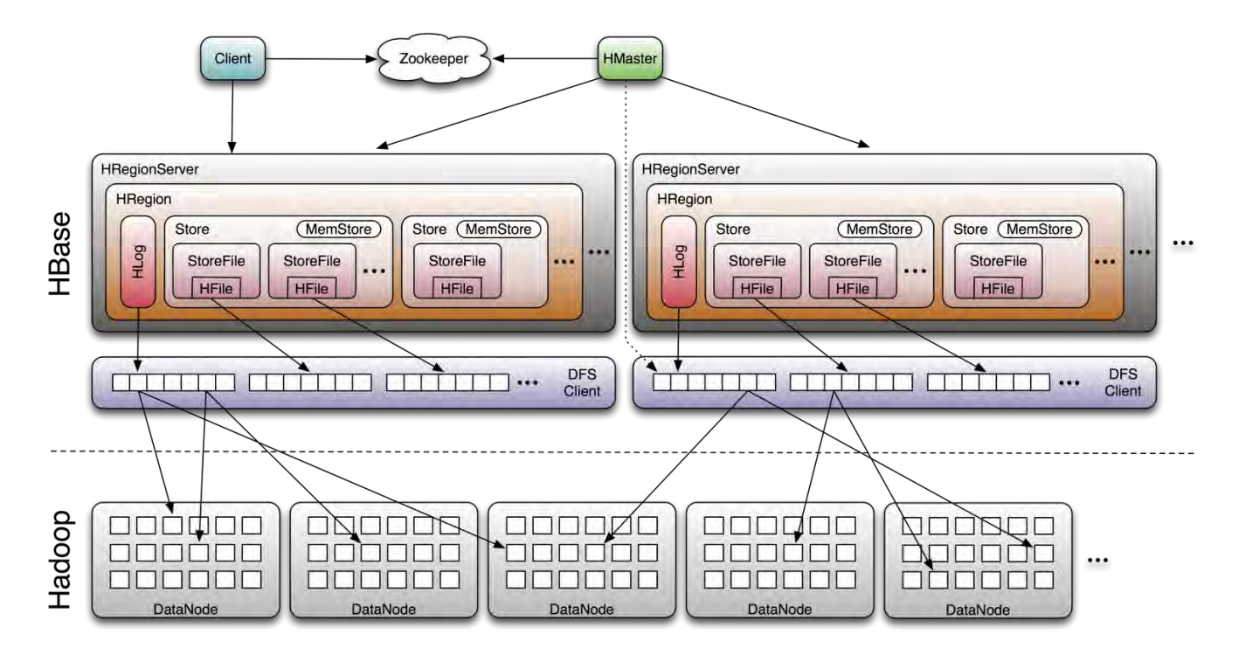


Abb. 14: HBase Architektur[75]

Wie in Abb. 14 zu sehen ist, beruht HBase auf dem „Verteilten System“ Hadoop, welches zur Speicherung der Daten verwendet wird.[76] Bei Hadoop handelt es sich um ein Framework, welches die verteilte Verarbeitung von großen Datenmengen in Clustern erlaubt. Somit kann HBase noch auf die grundlegenden Funktionen von Hadoop zurückgreifen, wie beispielsweise:[77]

- Hadoop MapReduce: Ein Framework zur parallelen Verarbeitung großer Datensätze
- Hadoop Yarn: Ein Framework zur Ressourcenverwaltung von Clustern und einer Optimierung von Scheduling-Funktionen.
- Hadoop Distributed File System (HDFS): Ein verteiltes Dateisystem, welches Zugriff auf Anwendungsdaten ermöglicht.
- Hadoop Common: Ein Dienst, welcher andere Hadoop Module unterstützt.

[74] Vgl. Edelich, S. u.a. (2010), S.55 f.
[75] Enthalten in: George, L. (2009)
[76] Vgl. Apache HBase (2014)
[77] Vgl. Apache Hadoop (2014)

Vor- und Nachteile

HBase ermöglicht eine kostengünstige Skalierung, da bereits durch Standardsoftware ein performantes Datenbanksystem aufgesetzt werden kann. Weiterhin bietet HBase vielfältige Schnitttstellen zur Anbindung an eigene Anwendungen an, wie z.B. Java-API. Durch das Map/Reduce-Framework können die Anfragen parallelisierte, effizient und effektiv ausgeführt werden. Allerdings erweist sich die Wartung, Optimierung und das Aufsetzen eines Clusters als sehr schwierig und komplex. Außerdem fehlt es an einer integrierten Replikation auf der Datenbankebene, da dies bisher nur auf Dateiebene stattfindet.[78]

4.2.3 Accumulo

	Apache Accumulo
Entwickler	US National Security Agency (NSA); Apache Software Foundation (seit 2011)
Letzte Version	Accumulo 1.6.1 (Release 24.09.2014)
Plattform	plattformunabhängig
Lizenz	Open-Source (Version 2 Apache Lizenz)
Einsatz	NSA (?)
Implementierung	Java

Abb. 15: Steckbrief Accumulo

Accumulo beruht auf dem Dateisystem Hadoop[79], dem Konfigurationsdienst ZooKeeper[80] und dem Thrift Framework[81], wodurch es möglich ist, erfolgreich einen Service zwischen zwei unterschiedlichen Programmiersprachen zu bauen. Wie die meisten NoSQL Datenbanken, hat Accumulo Googles BigTable als Vorbild. Die Entwicklung von Accumulo begann 2008 durch die NSA und trat der Apache Community 2011 bei.[82]

Funktionen

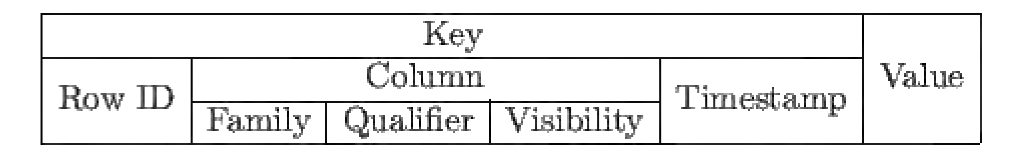


Abb. 16: Accumulo Datenmodell[83]

[78] Vgl. Edelich, S. u.a. (2010), S. 68 f.
[79] Vgl. dazu ausführlich: Apache Hadoop (2014)
[80] Vgl. dazu ausführlich: Apache ZooKeeper (2014)
[81] Vgl. dazu ausführlich: Apache Thrift (2014)
[82] Vgl. Apache Accumulo (2014a)
[83] Enthalten in: Apache Accumulo (2014b)

Auch Accumulo besteht, wie bei HBase und Cassandra auch, aus Key-/Value-Paaren (siehe Abb. 16), welche allerdings durch einen wesentlichen Punkt erweitert wurden. Die Daten bestehen aus Byte-Arrays, mit Ausnahme des Timestamps, welcher vom Typ Long ist.[84]

Im Folgenden werden einige Merkmale näher erläutert:[85]

- Row ID: Diese gruppiert mehrere Key-/Value-Paare in eine logische Reihe.
- Column Family: siehe Kapitel 4.1.
- Column Qualifier: Unterscheidet individuelle Columns in einer Column Family, da in Accumulo ein Key direkt einem Value zugeordnet ist, können dadurch einzigartige Keys für die Lagerung mehrerer Values in einer Column Family erzeugt werden.
- Column Visibility: Dies ist das entscheidende Merkmal von Accumulo, welches von den Accumulo Entwicklern dem ursprünglichen BigTable Modell hinzugefügt wurde. Durch diese Eigenschaft ist es möglich, Key-/Value-Paare nach bestimmten Zugangsberechtigungen zu filtern. Somit können in einer Tabelle, mehrere Daten mit unterschiedlichen Sicherheitsanforderungen gespeichert werden und jeder Benutzer sieht nur die Daten, für welche er berechtigt wurde. Man spricht hierbei von der cell-level access control.

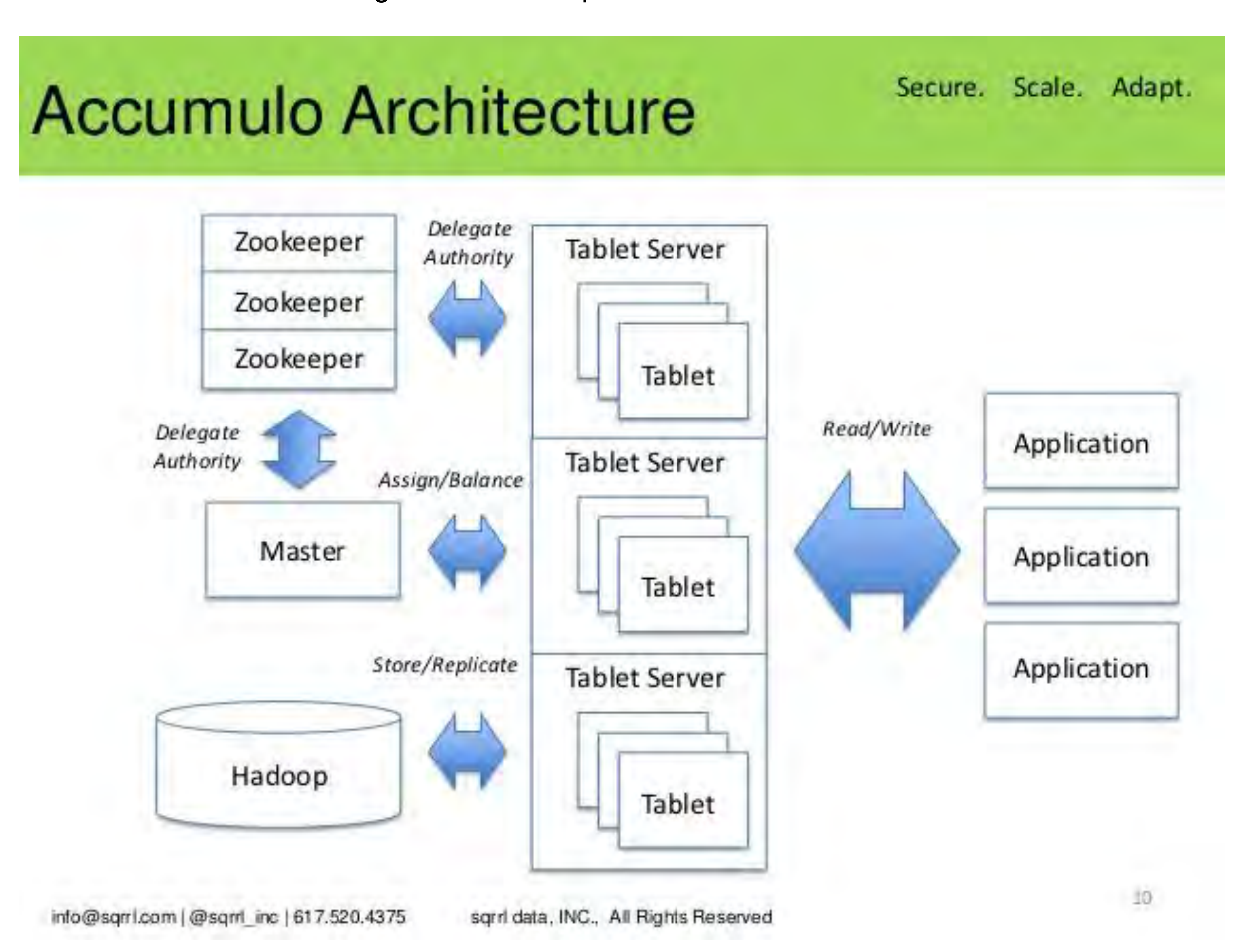


Abb. 17: Accumulo Architektur[86]

[84] Vgl. Apache Accumulo (2014b)
[85] Vgl. Safari (o. J.)

In Abb. 17 kann ein beispielhafter Aufbau einer Accumulo Architektur betrachtet werden. ZooKeeper ist verantwortlich für die Konfiguration. Die Tablet Server bestehen aus einer Teilmenge von Tablets (Partitionen von Tabellen) und nehmen die Anfragen der Anwendung entgegen. Der Master erkennt und reagiert auf Fehler der Tablet Server. Außerdem dient er als LoadBalancer und ist zuständig für die Erstellung, Änderung und das Löschen von Tabellen. Hadoop dient zur Speicherung der Daten.[87]

Vorteile und Nachteil

Das herausragende Merkmal bei Accumulo, ist die Sicherheit auf Zellebene. Somit eignet es sich speziell für Unternehmen, welche komplexe Sicherheitsrichtlinien haben. Weiterhin wurde bei Accumulo viel Wert darauf gelegt, das System durch zusätzliche Komponenten und Funktionen zu erweitern. Dies ist aber auch ein entscheidender Nachteil von Accumulo, da dadurch ein sehr komplexes System entsteht, wodurch die Handhabung schwierig wird.

4.3 Kriterienkatalog

„Katalog von Kriterien, der als Grundlage für eine Entscheidung o.Ä. dient.“[88]

Die Definition des Dudens bietet eine konkrete Vorstellung, welchem Verwendungszweck ein Kriterienkatalog dient. Der Kriterienkatalog umfasst nur die im Kapitel 4.2 betrachteten NoSQL-Systeme. Im Rahmen dieser Ausarbeitung ermöglicht der Kriterienkatalog eine detaillierte Auflistung und Bewertung der betrachteten Merkmale, auf welche im Folgenden näher eingegangen wird.

Ein wichtiger Punkt ist die Verwendung unterschiedlicher **Programmiersprachen**, wodurch die Erstellung der Datenbank flexibler gestaltet wird. Die **Lizenz** stellt ein weiteres Kriterium für die Auswahl einer Datenbank dar, da eventuell kosten entstehen könnten. Unter **Typisierung** ist das Vordefinieren des Datentyps zu verstehen, wodurch bei der Abfrage eine Definierung des Datentyps entfällt. Durch **Sekundärinidizes** können einzelne Daten nochmal geordnet werden und diese somit schneller ausgelesen werden. **Stored Procedures** sind serverseitige Skripts, welche automatisch vom Server ausgeführt werden und nicht durch den Client initiiert werden müssen. Unter dem Punkt **Partitionierung** ist die Speicherung unterschiedlicher Daten auf unterschiedlichen Knoten zu sehen, wobei bei Sharding der Datenbestand in mehrere Teile aufgeteilt und von einer eigenen Serverinstanz verwaltet wird. **Replikation** bedeutet das Speichern von Daten auf mehreren Knoten zur Datensicherung. Das **MapReduce** Prinzip ist in Kapitel 3.1 beschrieben. **Konsistenz** beschreibt die Methode,

---

[86] Enthalten in: Fuchs, A. (2012), S.10
[87] Vgl. Apache Accumulo (2014b)
[88] Duden (o. J.)

wodurch die Gleichheit der Daten bei einer Replikation sichergestellt werden sollen. Dabei werden die Daten bei „Immediate Consistency“ nach jeder Änderung gespeichert und bei „Eventual Consistency“ wird die Konsistenz durch Algorithmen erst nach Beendigung des Schreibvorgangs sichergestellt. Durch **parallele Datenmanipulation** wird das gleichzeitige Ausführen mehrerer Anweisungen ermöglicht. Die **dauerhafte Speicherung** stellt sicher, dass eine einmal durchgeführte Änderung der Datenbank dauerhaft erhalten bleibt. Durch die **Zugriffskontrolle** werden die Nutzerberechtigungen auf die einzelnen Datenbankinhalte geregelt. Die **Handhabung/Bedienbarkeit** ist ein intuitiver Wert, welcher auf Erfahrungsberichten verschiedener Nutzer beruht. Die **Praxisrelevanz** ergibt sich aus Abb. 10 in Kapitel 4.2. Der letzte Punkt **Zugriffsmöglichkeiten** listet die Schnittstellen oder auch Application Programming Interface (API) der DBMS auf. Der **Bewertungsspielraum** gibt die Anzahl der zu erreichenden Punkte jedes Merkmals wieder, wobei die Bewertung der Systeme jeweils in Klammer zu sehen ist.

	Bewertungsspielraum		Cassandra	HBase	Accumulo
Allgemein		Datenbanktyp	Wide Column Store	Wide Column Store	Wide Column Store
		Implementierungssprache	Java	Java	Java
		Datenschema	Schemafrei	Schemafrei	Schemafrei
		Untersuchte Version	2.1.2	2.0.0	1.6.1
Bewertungskriterien	**max. 4 Punkte pro DB**	Verwendbare Programmiersprachen	C#, C++, Java, JavaScript, Perl, PHP, Python, Ruby, u.a. **(4)**	C, C#, C++, Java, Groovy, PHP, Python, Scala **(4)**	C, C#, C++, Java, JavaScript, Perl, PHP, Python, Ruby, u.a. **(4)**
	max. 1 Punkt pro DB	Lizenz	Open Source **(1)**	Open Source **(1)**	Open Source **(1)**
	max. 1 Punkt pro DB	Typisierung	Ja **(1)**	Nein **(0)**	Nein **(0)**
	max. 1 Punkt pro DB	Sekundärindizes	nur Gleichheitsabfragen **(0,5)**	Nein **(0)**	Ja **(1)**
	max. 1 Punkt pro DB	Stored Procedures	Nein **(0)**	Ja **(1)**	Iterator **(1)**
	max. 1 Punkt pro DB	Partitionierung	Sharding **(1)**	Sharding **(1)**	Sharding **(1)**
	max. 1 Punkt pro DB	Replikation	Ja **(1)**	Ja **(1)**	Ja **(1)**
	max. 1 Punkt pro DB	MapReduce	Ja **(1)**	Ja **(1)**	Ja **(1)**
	max. 1 Punkt pro DB	Konsistenz	Eventual Consistency, Immediate Consistency **(1)**	Immediate Consistency **(1)**	Immediate Consistency **(1)**
	max. 1 Punkt pro DB	Parallele Datenmanipulation	Ja **(1)**	Ja **(1)**	Ja **(1)**
	max. 1 Punkt pro DB	Dauerhafte Speicherung	Ja **(1)**	Ja **(1)**	Ja **(1)**
	max. 1 Punkt pro DB	Zugriffskontrolle	Zugriff auf einzelne Objekte **(0,5)**	Access Control Lists (ACL) **(0,5)**	Zellebene **(1)**
	Nur 1 Punkt zu vergeben	Handhabung/Bedienbarkeit	relativ einfach **(1)**	komplex **(0)**	ziemlich komplex **(0)**
	Nur 1 Punkt zu vergeben	Praxisrelevanz (solid IT Ranking)	1. **(1)**	2. **(0)**	3. **(0)**
		∑	**14**	**12,5**	**14**
	Zugriffsmöglichkeiten (max. 4 Punkte pro DB)	Java API		x **(1)**	x **(1)**
		REST		x **(1)**	x **(1)**
		C++ API			
		Thrift	x **(1)**	x **(1)**	x **(1)**
		Sonstige	CQL **(3)**		Accumulo Shell, JDBC, ODBC **(1)**
		∑	**4**	**3**	**4**
Bewertung	∑		**18**	**15,5**	**18**

Abb. 18 Kriterienkatalog[89]

[89] Vgl. DB-Engines (o. J. c) vgl. auch dazu: Apache Accumulo (2014c) vgl. auch dazu: Broos, M. (2014)

Der Kriterienkatalog, in Abb. 18, zeigt eine Punktgleichheit von Cassandra und Accumulo. Allerdings ist, in Sachen Funktionalität, Accumulo der Spitzenreiter. Allerdings hinkt Accumulo durch seine Komplexität den beiden anderen Systemen hinsichtlich Handhabung/Bedienbarkeit hinterher. In diesem Punkt ist Cassandra der Vorreiter. Sowohl Accumulo als auch Cassandra haben bei den Zugriffsmöglichkeiten die höchst mögliche Punktzahl erreicht. Bei Accumulo ist dies durch seine unterschiedlichen Schnittstellen ersichtlich, wobei Cassandra durch seine zwei Möglichkeiten relativ schwach aussieht. Allerdings bietet Cassandra durch CQL eine an SQL angelehnte Abfragesprache mit vielen Möglichkeiten an, was CQL drei Punkte einbringt.

5 Erstellung von Datenbankprototypen

5.1 Kurzanleitung zur Installation

Um die Apache Cassandra Datenbank auf einem Windowsbetriebssystem zu installieren, muss zuerst die Umgebung eingerichtet werden. Es wird die Java Runtime Environment (JRE) benötigt. Nachdem diese installiert wurde muss, wenn noch nicht durch die Installation geschehen, die Umgebungsvariable initialisiert werden. Nach erfolgreichem Initialisieren der Umgebungsvariablen, kann die Cassandra Datenbank heruntergeladen werden. Auch hier muss die Umgebungsvariable initialisiert werden. Cassandra wird über das Kommandozeilenprogramm (CMD) gestartet, indem man die „cassandra.bat" Datei ausführt. Um nun die Anfragesprache Continuous Query Language (CQL) zu nutzen, wird zusätzlich Python benötigt.

Das Einlesen der Datensätze erfolgt durch den Java Quellcode (siehe Anhang 2). Dieser Quellcode wird mit Hilfe des Entwicklertools Eclipse erstellt. Hierzu wird zusätzlich das Java Development Kit (JDK) benötigt. Die benötigten Bibliotheken, um eine Verbindung zur Cassandra Datenbank aufzubauen, werden über das Tool „Maven" importiert.

5.2 Aufbau der Datenbankprototypen

Im Rahmen unseres Projekts wurden zwei Datenbanken erstellt. Einerseits eine Cassandra Datenbank als Vertreter der NoSQL Wide-Column-Datenbanken und andererseits eine MySQL Datenbank, welche die relationalen Datenbanken repräsentiert und als Vergleich herangezogen wird. Cassandra wurde gewählt, da es auf Grund der einfachen Installation gerade für Anfänger geeignet sein soll.

Die Datenbankprototypen sollen Sensormessungen von 22 Autos - unterschiedlicher Hersteller und Serien - erfassen. Dabei soll der genaue Ort und das Datum der Messung festgehalten werden. Es werden vier Sensoren erfasst:

1. **Benzinsensor:** Dieser überprüft den Benzinstand.
2. **Motorsensor:** Es erfolgt eine Überprüfung der Motortemperatur.
3. **Reifensensor:** Dieser überprüft sowohl den Reifendruck als auch die Reifentemperatur.
4. **Scheinwerfersensor:** Es wird aufgezeichnet, welche Scheinwerfer aktiv/inaktiv sind (Nebelscheinwerfer, Fernlicht, Blinker, Nebelschlussleuchte).

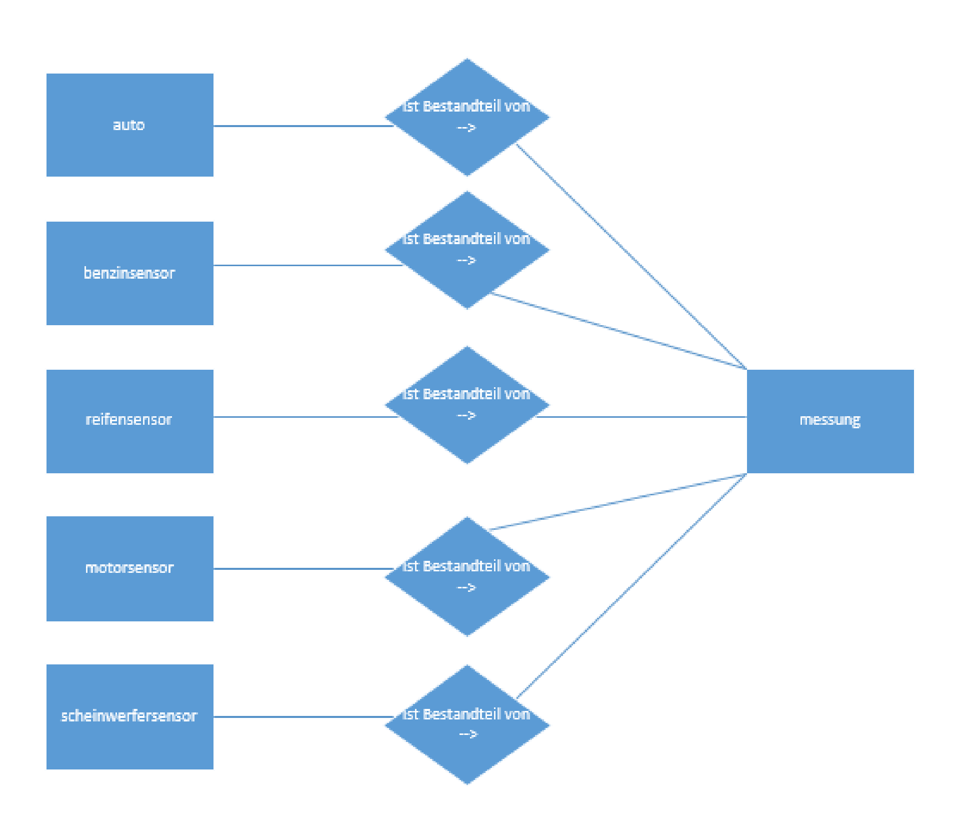


Abb. 19: ER-Diagramm MySQL

Das ER-Diagramm in Abb. 19 bildet die erste Basis für die MySQL Datenbank und zeigt die grundlegenden Beziehungen zwischen den einzelnen Tabellen. Zusätzlich zu den vier Sensoren, muss noch eine Tabelle mit den Autotypen („auto“) und eine Tabelle, in der die Messungen („messung“) zu einem bestimmten Zeitpunkt gespeichert werden, erzeugt werden.

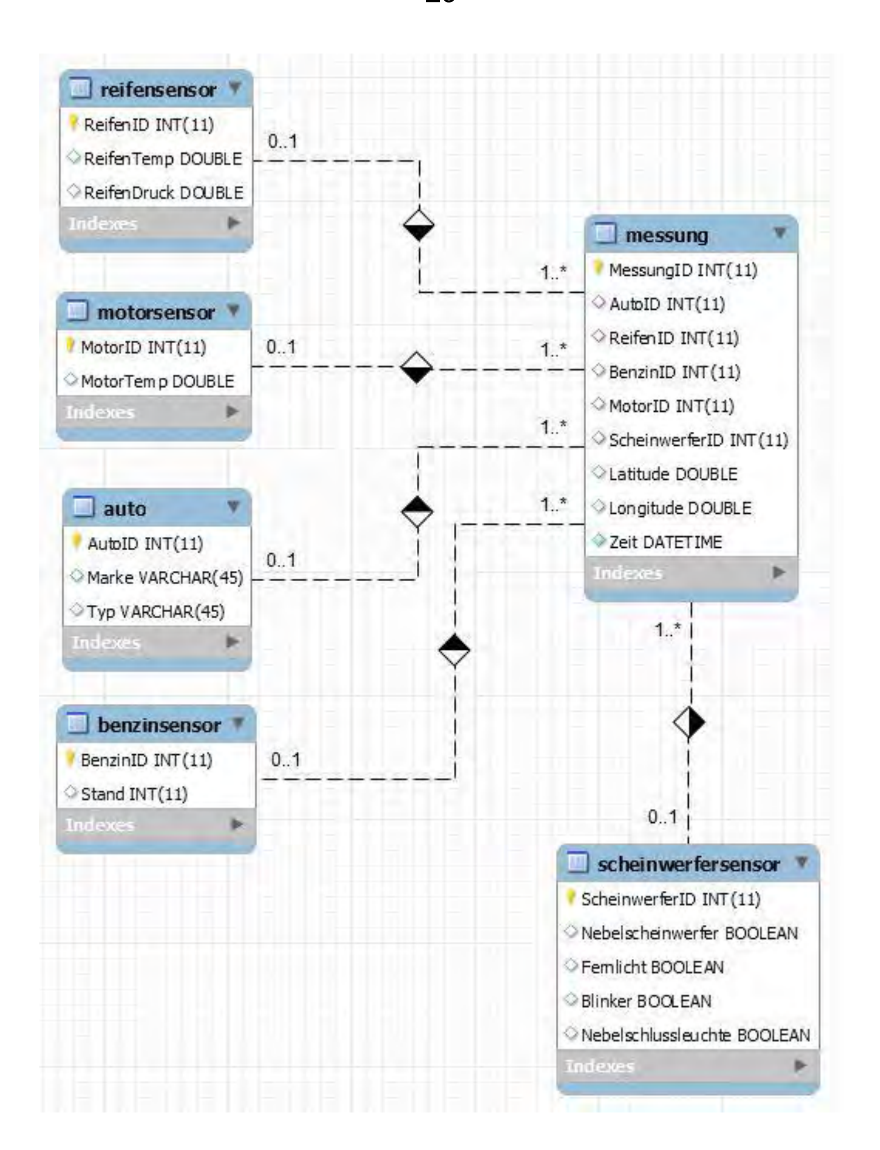
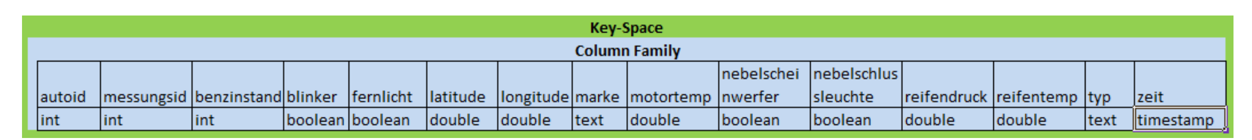


Abb. 20: Datenstruktur MySQL

In Abb. 20 ist die komplette Datenstruktur der MySQL Datenbank abgebildet. Die gelb markierten Attribute kennzeichnen die Primärschlüsselattribute der einzelnen Tabellen. Die Tabelle „messung" verweist durch die fünf Fremdschlüsselattribute („AutoID", „ReifenID"; „BenzinID", „MotorID", „ScheinwerferID") auf die anderen Tabellen.



Key-Space														
Column Family														
autoid	messungsid	benzinstand	blinker	fernlicht	latitude	longitude	marke	motortemp	nebelscheinwerfer	nebelschlusleuchte	reifendruck	reifentemp	typ	zeit
int	int	int	boolean	boolean	double	double	text	double	boolean	boolean	double	double	text	timestamp

Abb. 21: Cassandra Datenstruktur

Bei Cassandra können alle Attribute in einer Column Family gespeichert werden (siehe **Fehler! Verweisquelle konnte nicht gefunden werden.**). Es wird dementsprechend alles in einer Tabelle gespeichert, wobei jede Zeile alle notwendigen Informationen enthält. Bei der Cassandra Datenstruktur in Abb. 21 ist kein Primary Key hervorgehoben. Dies liegt daran,

dass es bei der Definierung eines einzelnen Primary Keys zu eingeschränkten Abfragemöglichkeiten kommt. Auf Grund dessen, mussten alle Attribute als zusammengesetzter Primary Key definiert werden (siehe Anhang 3).

Ein direktes Einlesen der Daten aus dem Java-Quellcode erzeugte, sowohl bei Cassandra als auch bei MySQL, erhebliche Performanceverluste. Deshalb wurde zur Verbesserung der Laufzeit Comma-Separated Values (CSV)-Dateien zur Generierung der Datenbankbestände erzeugt. (siehe Anhang 1).

Anschließen wurden die CSV-Dateien in die Datenbanken eingelesen (siehe Anhang 2).

6 Vergleich zwischen Wide-Column- und relationalen Datenbanken

6.1 Testsysteme

Um die späteren Ergebnisse zu verifizieren, erfolgt in erster Instanz eine Beschreibung der verwendeten Testumgebung:

- Betriebssystem: Windows 7
- Systemtyp: 32 Bit-Betriebssystem
- Prozessor: Intel® Core™ Duo CPU T8100 @2.10GHz
- Arbeitsspeicher: 2GB
- Grafikkarte: Mobile Intel 965 Express Chipset Family

Apache Cassandra:

- Cassandra 2.1.2
- CQL – Version: 3.2.0

MySQL:

- Workbench: 6.2
- MySQL Server: 5.6.21

Da es sich um eine ressourcenbeschränkte, lokale Umgebung handelt, konnte kein „Verteiltes System“ aufgebaut werden. Somit ist das Testen der horizontalen Skalierung nicht möglich. Dies widerspricht im wesentlichem dem Grundsatz der NoSQL-Datenbanken. Allerdings sollte eine generelle Unterscheidung bezüglich Abfragemöglichkeiten und –zeiten erreichbar sein.

6.2 Vergleichskriterien

Bevor eine Betrachtung der Abfrageanweisung erfolgt, müssen die Daten eingelesen werden. Dies stellt ein weiteres wichtiges Vergleichskriterium der beiden Datenbankprototypen dar.

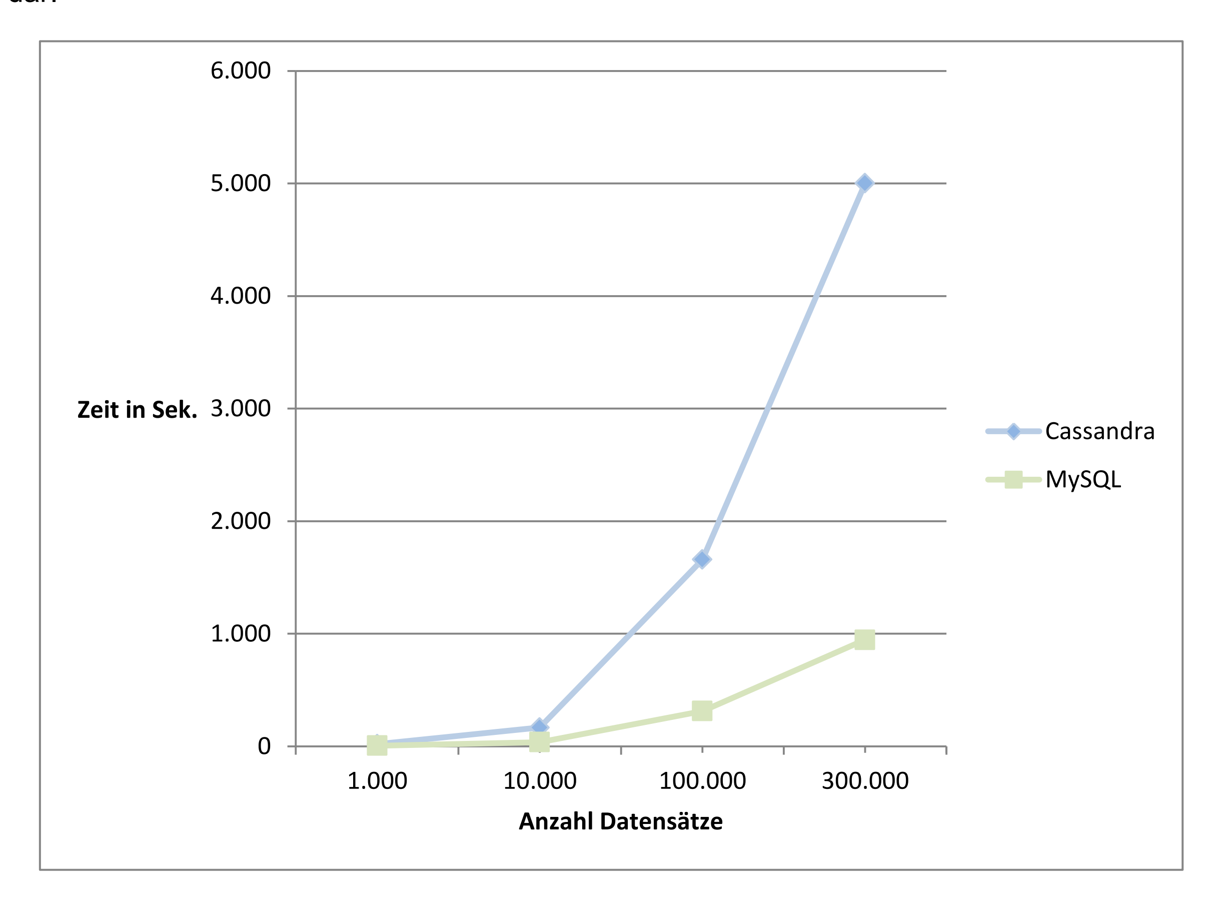


Abb. 22: Datenimport

Cassandra benötigt bei steigender Anzahl an Datensätzen deutlich mehr Zeit als MySQL (siehe Abb. 22). Bei 1000 Datensätzen zeichnet sich keine der beiden Datenbanktypen durch schnelleres Einlesen aus. Ab einer Menge von 10.000 Datensätzen wird Cassandra zunehmend schlechter. Dieser Unterschied wird bei 300.000 eingelesenen Daten sehr deutlich.

Um bei den nachfolgenden Abfragen deutliche Aussagen zu treffen, werden 1.000.000 Datensätze eingelesen. Dabei zeigt sich, dass Cassandra, bei diesem Umfang und dem zugrundeliegenden System, deutliche Probleme aufweist. Es kommt zu „TimeOuts“. Wohingegen es bei MySQL keine Probleme gibt. Dadurch mussten die Daten für Cassandra schrittweise eingelesen werden.

Syntax MySQL	MySQL	Syntax Cassandra	Cassandra
Select ReifenTemp From Reifensensor Where ReifenID IN (Select ReifenID From Messung where AutoID=10)	6959 (45317)	Select ReifenTemp From sensor.Messung Where AutoId=10 Allow Filtering	12128 (35875)
Select Motorsensor.MotorTemp, AutoID From Motorsensor JOIN Messung ON Motorsensor.MotorID=Messung.MotorID Where Latitude=8.104936 And Longitude=2.701645	2169 (1)	Select MotorTemp,AutoID From sensor.Messung Where Latitude=49.295145 And Longitude=16.431715 Allow Filtering	2461 (1)
Select auto.AutoIdD, Marke, Stand, Typ From Auto, Benzinsensor, Messung Where Auto.AutoID=Messung. AutoID and Benzinsensor.BenzinId =Messung.BenzinId	109426 (1000000)	Select AutoID,Marke,Benzinstand,Typ From sensor.Messung Allow Filtering	108508 (1000000)
Select Stand From Benzinsensor JOIN Messung ON Benzinsensor.BenzinID=Messung.BenzinID	21396 (1000000)	Select Benzinstand From sensor.Messung Allow Filtering	39366 (1000000)
Select Auto.AutoID, Marke, Typ From Auto JOIN Messung ON Auto.AutoID=Messung.AutoID Where MessungID IN (1, 500000, 1000000)	188 (3)	Select AutoID, Marke, Typ, MotorTemp From sensor.Messung Where MessungsID IN (1, 500000,1000000) Allow Filtering	78 (3)

Abb. 23: Select-Statements inklusive Laufzeiten in Millisekunden

Die Abb. 23 zeigt die Syntax der verwendeten Select-Statements, sowohl in MySQL als auch Cassandra. Ein beispielhafter Java Quellcode für das Abfragen der Daten befindet sich im Anhang 4. Die Abfragezeiten befinden sich in der nachfolgenden Spalte, wobei die Anzahl der ausgegebenen Datensätze in Klammer gesetzt ist. Es fällt auf, dass bei der ersten Abfrage eine unterschiedliche Anzahl an Datensätzen ausgegeben wurde. Dies ergibt sich aus dem Umstand, dass unterschiedliche Datensätze eingespielt wurden. Das ist jedoch bei einem Datenbestand von 1.000.000 Datensätzen zu vernachlässigen, da bei einer so großen Datenmenge, mit einer Gleichverteilung zu rechnen ist. Die MySQL Statements sind auf

Grund der Datenstruktur komplexer gestaltet, wohingegen die Cassandra Statements übersichtlicher sind (siehe Abb. 23). Das erste Statement zeigt deutlich die Vorteile von MySQL, bei einer Anfrage innerhalb einer einzelnen Tabelle. Bei Abfragen nach bestimmten Werten, bei welcher MySQL erneut mehrere Tabellen mit einbeziehen muss, ist Cassandra im Vorteil (siehe letzte Zeile Abb. 23). Cassandra mangelt es jedoch an wesentlichen Abfragemöglichkeiten. So ist es nicht möglich Mittelwerte- und Summen- Statements zu bilden. Außerdem mussten alle Attribute bei Cassandra als Primary Keys definiert werden, um beispielsweise IN Operation auszuführen. Außerdem lassen sich DESC und ASC Sortierungen nur über den Partition Key (der zuerst definierte Primary Key[90]) ausführen.

[90] Vgl. Datastax (2014)

7 Schlussbetrachtung

Die Analyse der Ergebnisse zeigt gravierende Unterschiede zwischen Cassandra und MySQL. Es hat sich gezeigt, dass MySQL sowohl bei Einlesezeiten als auch bei konkreten Statements bedeutend schneller agieren kann. Bei den Abfragen hat sich gezeigt, dass die Select-Statements von MySQL viel komplexer sind als bei Cassandra. Dies ist auf das ACID-Prinzip bei relationalen Datenbanken zurückzuführen.

Dennoch zeigt sich, dass die meisten Antwortzeiten bei MySQL deutlich schneller sind. Dies lässt sich allerdings nur für die eingesetzte Testumgebung sagen. Es kann davon ausgegangen werden, dass bei einem komplexeren Testszenario, welches eine Vielzahl von Join-Operationen zulässt, MySQL seinen Vorsprung verlieren würde. Dies ließ sich in dieser Arbeit aber nicht genauer verifizieren.

Um eine qualitative Aussage darüber zu treffen, ob und wann Cassandra eine mögliche Alternative darstellt, müsste eine komplette „Verteilte Testumgebung" aufgebaut werden. Es lässt sich allerding vermuten, dass Cassandra auf Grund ihrer Ausrichtung als „Verteiltes System", deutlich bessere Ergebnisse liefern würde.

Darüber hinaus basiert Cassandra und andere Wide-Column-Datenbanken auf der horizontalen Skalierbarkeit. Dies führt dazu, dass kostengünstige performancesteigernde Hardwarekapazitäten hinzugefügt werden kann. Somit eignen sich die Wide-Column Datenbank besonders für den Einsatz riesiger Datenmengen. Ein weiterer Vorteil dieser Datenbanken liegt in der Fähigkeit, Daten unterschiedlichen Strukturformats gemeinsam abspeichern zu können.

Deshalb kann durch diese Ausarbeitung nachvollzogen werden, warum Google und Amazon bei Suchanfragen Wide-Column-Datenbanken verwenden. Hier treten unterschiedlichste Daten auf, die für ihre Kunden schnellstmöglich ausgewertet werden müssen, ohne auf das ACID-Prinzip angewiesen zu sein.

Es lässt sich sagen, dass sowohl das ACID als auch des BASE Prinzip ihre Daseinsberechtigung haben. Je nach Einsatzszenario ist es sinnvoll auf das ein oder andere Prinzip zu setzen. Die jeweiligen Vorteile des einen Systems sind die Nachteile des anderen.

Schlussendlich lässt sich sagen, dass sich Cassandra und andere Wide-Column-Datenbanken noch in der Entwicklung befinden. Dies spiegelt sich vor allem in den unausgereiften Abfragemöglichkeiten und beschränkten Einsatzgebieten wider.

Anhang

Anhang 1: Cassandra-CSV

Im Folgenden ist der Quellcode für die Erstellung der Cassandra csv Datei zu sehen, welche für die Erstellung der Datensätze verwendet wurde:

```
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Calendar;
import java.util.Date;
import java.util.Random;

public class CassandraCSV {

public static void main(String[] args) {

		// Schleifenzähler 1 <= zaehler
		int zaehler = 1000000;

		// Autodaten
		String Marke;
		String Typ;
		int AutoID;

		// Für Reifensensor
		double reifenTempStart = 0;
		double reifenTempEnde = 60;
		double reifenDruckStart = 0;
		double reifenDruckEnde = 2.5;

		// Für Motorsensor
		double motorStart = 0;
		double motorEnde = 120;

		// Automarken und Typen
		String[] automarke = { "BMW", "Audi", "VW", "Mercedes" };
		String[] autotypBMW = { "3er", "4er", "5er", "6er", "7er" };
		String[] autotypAudi = { "A1", "A3", "A4", "A5", "A6", "A7", "A8" };
		String[] autotypVW = { "Golf", "Polo", "Passat", "Touran", "Sharan"
};
		String[] autotypMercedes = { "A-Klasse", "B-Klasse", "C-Klasse",
```

```
                "E-Klasse", "S-Klasse" };

        // Für Longitude und Latitude
        double loStart = 0;
        double loEnde = 20;
        double laStart = 0;
        double laEnde = 60;

        // Für die Trennung der einzelnen Daten in der .csv Datei
        String s = ",";

        // Autosensor .csv Datei erstellen
        try {
            BufferedWriter bufferedwriter = new BufferedWriter(
                    new FileWriter(

    "C:/Users/Arbeit/CloudStation/Projektarbeit/Projekt/Cassandra/CassandraCSV.c
sv"));

            // Anzahl der zu erstellenden Daten definieren
            for (int i = 1; i <= zaehler; i++) {

                //MessungsID's erstellen = Primärschlüssel
                int MessungsID = i;

                // Autodatendaten
                Random auto_r = new Random();
                String r_automarke = automarke[auto_r.nextInt(4)];

                if (r_automarke == "BMW") {
                    Marke = "BMW";
                    Typ = autotypBMW[auto_r.nextInt(5)];
                    if (Typ == "3er") {
                        AutoID = 1;
                    } else if (Typ == "4er") {
                        AutoID = 2;
                    } else if (Typ == "5er") {
                        AutoID = 3;
                    } else if (Typ == "6er") {
                        AutoID = 4;
                    } else {
```

```
                AutoID = 5;
        }

} else if (r_automarke == "Audi") {
        Marke = "Audi";
        Typ = autotypAudi[auto_r.nextInt(7)];
        if (Typ == "A1") {
                AutoID = 6;
        } else if (Typ == "A3") {
                AutoID = 7;
        } else if (Typ == "A4") {
                AutoID = 8;
        } else if (Typ == "A5") {
                AutoID = 9;
        } else if (Typ == "A6") {
                AutoID = 10;
        } else if (Typ == "A7") {
                AutoID = 11;
        } else {
                AutoID = 12;
        }

} else if (r_automarke == "VW") {
        Marke = "VW";
        Typ = autotypVW[auto_r.nextInt(5)];
        if (Typ == "Golf") {
                AutoID = 13;
        } else if (Typ == "Polo") {
                AutoID = 14;
        } else if (Typ == "Passat") {
                AutoID = 15;
        } else if (Typ == "Touran") {
                AutoID = 16;
        } else {
                AutoID = 17;
        }

} else {
        Marke = "Mercedes";
        Typ = autotypMercedes[auto_r.nextInt(5)];
```

```
                    if (Typ == "A-Klasse") {
                        AutoID = 18;
                    } else if (Typ == "B-Klasse") {
                        AutoID = 19;
                    } else if (Typ == "C-Klasse") {
                        AutoID = 20;
                    } else if (Typ == "E-Klasse") {
                        AutoID = 21;
                    } else {
                        AutoID = 22;
                    }
                }

                // Reifensensordaten
        double reifen_r = new Random().nextDouble();
        double reifenZahl1 = reifenDruckStart
                + (reifen_r * (reifenDruckEnde - reifenDruckStart));
        double ReifenDruck = Math.round(100.0 * reifenZahl1) / 100.0;
        double reifenZahl2 = reifenTempStart
                    + (reifen_r * (reifenTempEnde - reifenTempStart));
        double ReifenTemp = Math.round(100.0 * reifenZahl2) / 100.0;

                // Motorsensordaten
        double motor_r = new Random().nextDouble();
        double motorZahl1 = motorStart
                        + (motor_r * (motorEnde - motorStart));
        double MotorTemp = Math.round(100.0 * motorZahl1) / 100.0;

                // Benzinsensordaten
        java.util.Random benzin_r = new java.util.Random();
        int Benzinstand = 33 + benzin_r.nextInt(207); // 33 <= ohm < 240

                // Scheinwerfersensordaten
        java.util.Random scheinwerfer_r = new java.util.Random();
        boolean Blinker = scheinwerfer_r.nextBoolean();
        boolean Fernlicht = scheinwerfer_r.nextBoolean();
        boolean Nebelscheinwerfer = scheinwerfer_r.nextBoolean();
        boolean Nebelschlussleuchte = scheinwerfer_r.nextBoolean();

                // Für Longitude und Latitude
```

```
double lo_la_r = new Random().nextDouble();
double lo_la = laStart + (lo_la_r * (laEnde - laStart));
double Latitude = Math.round(1000000.0 * lo_la) / 1000000.0;
double lo_la2 = loStart + (lo_la_r * (loEnde - loStart));
double Longitude = Math.round(1000000.0 * lo_la2) / 1000000.0;

// Timestamp generieren und für jeden Datensatz um eins erhöhen
Date Zeit = new Date();
            Calendar c = Calendar.getInstance();
            c.setTime(Zeit);
            c.add(Calendar.DATE, i);
            Zeit = c.getTime();

      // Erstellte Daten in die .csv Datei schreiben
      bufferedwriter.write(String.valueOf(MessungsID));
      bufferedwriter.write(s);
      bufferedwriter.write(String.valueOf(AutoID));
      bufferedwriter.write(s);
      bufferedwriter.write(String.valueOf(Marke));
      bufferedwriter.write(s);
      bufferedwriter.write(String.valueOf(Typ));
      bufferedwriter.write(s);
      bufferedwriter.write(String.valueOf(ReifenTemp));
      bufferedwriter.write(s);
      bufferedwriter.write(String.valueOf(ReifenDruck));
      bufferedwriter.write(s);
      bufferedwriter.write(String.valueOf(MotorTemp));
      bufferedwriter.write(s);
      bufferedwriter.write(String.valueOf(Benzinstand));
      bufferedwriter.write(s);
      bufferedwriter.write(String.valueOf(Nebelscheinwerfer));
      bufferedwriter.write(s);
      bufferedwriter.write(String.valueOf(Fernlicht));
      bufferedwriter.write(s);
      bufferedwriter.write(String.valueOf(Blinker));
      bufferedwriter.write(s);
      bufferedwriter.write(String.valueOf(Nebelschlussleuchte));
      bufferedwriter.write(s);
      bufferedwriter.write(String.valueOf(Latitude));
      bufferedwriter.write(s);
```

```
                bufferedwriter.write(String.valueOf(Longitude));
                bufferedwriter.write(s);
                bufferedwriter.write(String.valueOf(Zeit));
                bufferedwriter.newLine();
                }
                System.out.println("Cassandra CSV-Datei wurden erstellt.");
                bufferedwriter.close();
            } catch (IOException e) {
                System.out.println("Fehler: " + e.toString());
            }

        }

    }
```

Anhang 2: Cassandra-Import-CSV

Im Folgenden ist der Quellcode für den Import der Cassandra csv Datei in die Cassandra Datenbank zu sehen:

```
package com.example.cassandra;

import java.io.FileReader;

import com.datastax.driver.core.BoundStatement;
import com.datastax.driver.core.Cluster;
import com.datastax.driver.core.PreparedStatement;
import com.datastax.driver.core.Session;
import com.opencsv.CSVReader;

public class CassandraImportCSV {

	//Variablen für die Cassandra-Datenbankverbidnung erstellen
	static Cluster cluster;
	static Session session;

	public static void main(String[] args) {

		//Methode ausführen
		readCsv();
	}

	private static void readCsv() {
		long zeit_start = System.currentTimeMillis();


		// CassandraCSV.csv Daten einlesen
		try (CSVReader reader = new CSVReader(
					new FileReader(

	"C:/Users/Arbeit/CloudStation/Projektarbeit/Projekt/Cassandra/CassandraCSV.c
sv"),
					',');) {

			// Verbindungsaufbau zu Cassandra
		cluster = Cluster.builder().addContactPoint("127.0.0.1").build();
```

```
        session = cluster.connect("sensor");

            // Insert Befehl erstellen und ausführen
String strCQL = "Insert INTO sensor.messung (MessungsID, AutoID, Marke, Typ, Rei-
fentemp, Reifendruck, Motortemp, Benzinstand, Nebelscheinwerfer, Fernlicht, Blin-
ker, Nebelschlussleuchte, Latitude, Longitude, Zeit) VALUES (?, ?, ?, ?, ?, ?, ?,
?, ?, ?, ?, ?, ?, ?, ?)";
        PreparedStatement preparedStatement = session.prepare(strCQL);

            // Daten in ein String-Array einlesen und in den entsprechenden
            // Datentyp parsen.
        String[] rowData = null;

        while ((rowData = reader.readNext()) != null) {

                int MessungsID = Integer.parseInt(rowData[0]);
                int AutoID = Integer.parseInt(rowData[1]);
                String Marke = rowData[2];
                String Typ = rowData[3];
                double Reifentemp = Double.parseDouble(rowData[4]);
                double Reifendruck = Double.parseDouble(rowData[5]);
                double Motortemp = Double.parseDouble(rowData[6]);
                int Benzinstand = Integer.parseInt(rowData[7]);
                String Nebelscheinwerfer = rowData[8];
                String Fernlicht = rowData[9];
                String Blinker = rowData[10];
                String Nebelschlussleuchte = rowData[11];
                double Latitude = Double.parseDouble(rowData[12]);
                double Longitude = Double.parseDouble(rowData[13]);
                String Zeit = rowData[14];




                // Daten werden der Datenbank übergeben
BoundStatement boundStatement = preparedStatement
    .bind(MessungsID, AutoID, Marke, Typ, Reifentemp, Reifendruck, Motortemp,
    Benzinstand, Nebelscheinwerfer, Fernlicht, Blinker, Nebelschlussleuchte, La-
    titude, Longitude, Zeit);
                session.execute(boundStatement);
            }
            System.out.println("Messungsdaten erfolgreich eingelesen");
```

```
            } catch (Exception e) {
                e.printStackTrace();
            }

            System.out.println("Daten sind eingelesen.");

            long zeit_ende = System.currentTimeMillis();
            long messung = zeit_ende - zeit_start;
            System.out.println("Die Laufzeit beträgt: " + messung + "ms");
        }
    }
```

Anhang 3: Cassandra Create Statements

Die Abb. 1 und Abb. 2 stellen beide Möglichkeiten dar, um eine Cassandra Datenbank zu erzeugen. Der Unterschied liegt ausschließlich in der Definierung des Primary Keys. In Abb. 1 wird ein zusammengesetzter Primary Key verwendet, wogegen in Abb. 2 die „MessungsID" als Primary Key definiert ist. Diesbezüglich müssen in dem Create Statement in Abb. 2 noch Indexe für eine bessere Suchabfrage und um gewisse Suchoperatoren (z.B. „="-Abfragen) erzeugt werden. Zuerst wird ein Keyspace („Sensor") mit definierten Replikationseigenschaften erzeugt. Anschließend wird in diesem Keyspace eine Column-Family („Messung") erstellt und mit Attributen befüllt.

```
CREATE KEYSPACE Sensor WITH replication = {'class': 'SimpleStrategy', 'replication_factor' : 1};

USE Sensor;

CREATE TABLE Messung (
    MessungsID int,
    AutoID int,
    Marke text,
    Typ text,
    Reifentemp double,
    Reifendruck double,
    Motortemp double,
    Benzinstand int,
    Nebelscheinwerfer text,
    Fernlicht text,
    Blinker text,
    Nebelschlussleuchte text,
    Latitude double,
    Longitude double,
    Zeit text,
    PRIMARY KEY (MessungsID, AutoID, Marke, Typ, Reifentemp,
                Reifendruck, Motortemp, Benzinstand, Nebelscheinwerfer,
                Fernlicht, Blinker, Nebelschlussleuchte, Latitude, Longitude, Zeit));
```

Abb. 1: Create-Statement Cassandra mit Primary Key über alle Attribute

```
CREATE KEYSPACE Sensor WITH replication = {'class': 'SimpleStrategy', 'replication_factor' : 1};

USE Sensor;

CREATE TABLE Messung (
    MessungsID int,
    AutoID int,
    Marke text,
    Typ text,
    Reifentemp double,
    Reifendruck double,
    Motortemp double,
    Benzinstand int,
    Nebelscheinwerfer text,
    Fernlicht text,
    Blinker text,
    Nebelschlussleuchte text,
    Latitude double,
    Longitude double,
    Zeit text,
    PRIMARY KEY (MessungsID));

CREATE INDEX ON Sensor.Messung (Marke);
CREATE INDEX ON Sensor.Messung (Typ);
CREATE INDEX ON Sensor.Messung (Reifentemp);
CREATE INDEX ON Sensor.Messung (Reifendruck);
CREATE INDEX ON Sensor.Messung (Motortemp);
CREATE INDEX ON Sensor.Messung (Benzinstand);
CREATE INDEX ON Sensor.Messung (Nebelscheinwerfer);
CREATE INDEX ON Sensor.Messung (Fernlicht);
CREATE INDEX ON Sensor.Messung (Blinker);
CREATE INDEX ON Sensor.Messung (Nebelschlussleuchte);
CREATE INDEX ON Sensor.Messung (Latitude);
CREATE INDEX ON Sensor.Messung (Longitude);
CREATE INDEX ON Sensor.Messung (Zeit);
```

Abb. 2: Create-Statement Cassandra mit einem Primary Key

Anhang 4: Select- Statements Cassandra

```
package com.example.cassandra;

import com.datastax.driver.core.Cluster;
import com.datastax.driver.core.Host;
import com.datastax.driver.core.Metadata;
import com.datastax.driver.core.ResultSet;
import com.datastax.driver.core.Row;
import com.datastax.driver.core.Session;

public class SimpleClient {
	private Cluster cluster;
	private Session session;
	int zaehler = 0;

	//Verbindungsaufbau und Metadatenausgabe
	   public void connect(String node) {
	      cluster = Cluster.builder()
	            .addContactPoint(node).build();
	      Metadata metadata = cluster.getMetadata();
	      System.out.printf("Connected to cluster: %s\n",
	            metadata.getClusterName());
	      for ( Host host : metadata.getAllHosts() ) {
	         System.out.printf("Datatacenter: %s; Host: %s; Rack: %s\n",
	               host.getDatacenter(), host.getAddress(), host.getRack());
	      }
	      session = cluster.connect();
	   }

	   // Select-Abfrage
	   public void querySchema (){
			long zeit_start = System.currentTimeMillis();
			ResultSet results = session.execute("SELECT BenzinStand FROM sen-
sor.messung ALLOW FILTERING");

				for (Row row : results) {
					zaehler = zaehler + 1;

					/*System.out.println(row.getInt("AutoID"));
				System.out.println(row.getString("Marke"));
				System.out.println(row.getString("Typ"));
				System.out.println(row.getDouble("MotorTemp"));
				System.out.println(row.getDouble("ReifenTemp"));
				System.out.println(row.getDouble("ReifenDruck"));
				System.out.println(row.getInt("MessungsID"));
				System.out.println(row.getInt("ScheinwerferID"));
				System.out.println(row.getInt("Benzinstand"));
				System.out.println(row.getString("Nebelschlussleuchte"));*/

				}
				long zeit_ende = System.currentTimeMillis();
				long messung = zeit_ende - zeit_start;
				System.out.println("Die Laufzeit beträgt: " + messung + "ms");
				System.out.println("Zeilen: "+zaehler);
	   }
```

```
        public void close() {
            cluster.close();
        }

        public static void main(String[] args) {
            SimpleClient client = new SimpleClient();
            client.connect("127.0.0.1");
            client.querySchema();
            client.close();
        }

}
```

Quellenverzeichnisse

Literaturverzeichnis

Edlich, S. u. a. (2010): NoSQL, Einstieg in die Welt nichtrelationaler Web 2.0 Datenbanken, (Hrsg.: Edlich, S., Friedland, A., Hampe, J., Brauer, B.), 7. Auflage, München: Carl Hanser Verlag GmbH & Co. KG

Fielding, R. (2010): Architectural Styles and the Design of Networkbased Software Architectures, Irvine: University of California

Ngoc Ha, T. (2011): Charakteristika und Vergleich von SQL- und NoSQL-Datenbanken, Leipzig: Universität Leipzig

Sadalage, Promod J. / Fowler, Martin (2012): NoSQL distilled, A brief guide to the emerging world of polygot persistence, 1. Auflage, London: Addison-Wesley

Thurm, B. (2012): Einsatz von NoSQL-Datenbanksystemen für Geodaten, Dresden: Hochschule für Technik und Wirtschaft Dresden

Verzeichnis der Internet- und Intranet-Quellen

Apache Accumulo (2014a):	Accumulo, https://accumulo.apache.org/, Abruf: 19.12.2014
Apache Accumulo (2014b):	Apache Accumulo User Manual: Accumulo Design, http://accumulo.apache.org/user_manual_1.3-incubating/Accumulo_Design.html, Abruf: 19.12.2014
Apache Accumulo (2014c):	Apache Accumulo User Manual Version 1.6, http://accumulo.apache.org/1.6/accumulo_user_manual.html, Abruf: 08.01.2015
Apache Haadop (2014):	Welcome to Apache Hadoop, http://hadoop.apache.org/, Abruf: 19.12.2014
Apache HBase (2014):	Welcome to Apache Hbase, http://hbase.apache.org/, Abruf: 19.12.2014
Apache Thrift (2014):	Apache Thrift, http://thrift.apache.org/, Abruf: 19.12.2014
Apache ZooKeeper (2014):	Welcome to Apache ZooKeeper, http://zookeeper.apache.org/, Abruf: 19.12.2014
Broos, Marc (2014):	HBase versus Cassandra versus Accumulo, http://bigdata-guide.blogspot.de/2014/01/hbase-versus-cassandra-versus-accumulo.html, Abruf: 08.01.2015
Datastax (2014):	Documentation for CQL 1.2, http://www.datastax.com/documentation/cql/3.0/share/glo gloss/gloss_partition_key.html, Abruf: 20.01.2015
DB-Engines (o. J. a):	Berechnungssmethode der Wertung im DB-Engines Ranking, http://db-engines.com/de/ranking_definition, Abruf: 08.01.2014
DB-Engines (o. J. b):	Schemafreiheit, http://db-engines.com/de/article/Schemafreiheit, Abruf: 04.12.2014
DB-Engines (o. J. c):	Vergleich der Systemeigenschaften Cassandra vs. HBase, db-engines.com/de/system/Cassandra%3BHBase, Abruf: 08.01.2015

DB-Engines (2014a): DB-Engines Ranking von Document Stores, http://db-engines.com/de/ranking/document+store,
Abruf: 14.12.2014

DB-Engines (2014b): DB-Engines Ranking von Graph DBMS, http://db-engines.com/de/ranking/graph+dbms,
Abruf: 14.12.2014

DB-Engines (2014c): DB-Engines Ranking von Wide Column Stores, http://db-engines.com/de/ranking/wide+column+store,
Abruf: 14.12.2014

Duden (o. J.): Kriterienkatalog,
http://www.duden.de/rechtschreibung/Kriterienkatalog,
Abruf: 08.01.2015

FH Köln (2011): MVCC – Multiversion Concurrency Control,
http://wikis.gm.fh-koeln.de/wiki_db/Datenbanken/MVCC,
Abruf: 08.01.2015

FH Köln (2012): Representational State Transfer,
http://wikis.gm.fh-koeln.de/wiki_db/Datenbanken/REST,
Abruf: 08.01.2015

FH Köln (2014): Spaltenorientierte Datenbanksysteme, http://wikis.gm.fh-koeln.de/wiki_db/Datenbanken/SpaltenorientierteDatenbank,
Abruf: 08.01.2015

Fuchs, Adam (2012): sqrrl data INC., http://de.slideshare.net/ydn/hug-accumulo, Abruf: 08.01.2015

Georg, L. (2009): HBase Architecture 101 – Storage,
http://www.larsgeorge.com/2009/10/hbase-architecture-101-storage.html,
Abruf: 08.01.2015

Hahn Projects (o. J.): Distributed Traffic Data I,
http://www.hahnpro.com/distributed-traffic-data-i/,
Abruf: 08.01.2015

Klein, D. / Tran-Gia, P. /
Hartmann, M. (2013): Big Data,
http://www.gi.de/service/informatiklexikon/detailansicht/art

icle/big-data.html,
Abruf: 30.11.2014

MacFadden, G. (2013): 21 NoSQL Innovators to Look for in 2020, http://wikibon.org/wiki/v/21_NoSQL_Innovators_to_Look_for_in_2020,
Abruf: 15.12.2014

Niemann C. (2011): Big Data: Herausforderung für Datenbanken, http://www.zdnet.de/41558518/big-data-herausforderung-fuer-datenbanken,
Abruf: 30.11.2014

NoSQL-Database (o.J.): List of NoSQL Databases,
http://nosql-database.org,
Abruf: 26.11.2014

Pientka, F. (2011): Datenbanken: NOSQL im BigData-Land, http://www.it-daily.net/it-strategie/enterprise-it/5250-datenbanken-nosql-im-bigdata-land,
Abruf: 30.11.2014

Safar (o. J.): Accumulo,
https://www.safaribooksonline.com/library/view/accumulo/9781491947098/ch01.html, Abruf: 08.01.2015

Wolff, E. (2012): NoSQL – die neue Datenbankgeneration?,
http://www.computerwoche.de/a/nosql-die-neue-datenbankgeneration,1235662,
Abruf: 30.11.2014



Vergleich von Open Source Produkten zum Aufbau einer Private Cloud

Schriftliche Ausarbeitung
im Rahmen der Lehrveranstaltung „Integrationsseminar“
für das Kompetenzzentrum Open Source (KOS)

Vorgelegt von

Feline Bohn, Franziska Gross,
Mareike Häfner, Tobias Burkhardt,
Anja Beyerbach

am 26.01.2015

Fakultät Wirtschaft
Studiengang Wirtschaftsinformatik
WWI2012I

Inhaltsverzeichnis

Abkürzungsverzeichnis

API	Application Programming Interface (Programmierschnittstelle)
AWS	Amazon Web Services
AWS S3	Amazon Web Services – Simple Storage Service
BDSG	Bundesdatenschutzgesetz
CPU	Central Processing Unit (Hauptprozessor)
GUI	Graphical User Interface (Grafische Benutzeroberfläche)
IaaS	Infrastructure as a Service
LDAP	Lighweight Directory Access Protocol (Leichtgewichtiges Verzeichniszugriffsprotokoll)
NIST	National Institute of Standards and Technology
PaaS	Platform as a Service
RSA	Rivest, Shamir, Adleman, nach den Erfindern benannt
SaaS	Software as a Service
SLA	Service Level Agreement
UrhG	Urheberrechtsgesetz
VM	Virtuelle Maschine

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

1.1 Problemstellung

„Cloud Computing [is] the Fifth Generation of Computing. After Mainframe, Personal Computer, Client-Server Computing, and the Web."

(**Andreas von Bechtolsheim**, Mitbegründer Sun Microsystems)[1]

Die erste namentliche Nennung des *Cloud Computings* erfolgte 2006 auf einer Pressekonferenz des Unternehmens *Google*. Eric Schmidt – zu dem Zeitpunkt CEO von *Google* – bezeichnete Dienstleistungen zur Datenverwaltung und die Bereitstellung von IT-Architektur auf von mehreren Personen zugänglichen Servern als *Cloud*.[2] Seit der Prägung dieses Begriffs ist das Konzept des Cloud Computings enorm gewachsen und ist aus unserem Alltag – privat wie beruflich – nicht mehr wegzudenken.[3] Dokumentenbereitstellung – Bilderaustausch – Kommunikationssysteme – Back-Up-Lösungen – Quellcodeversionsverwaltung – Anwenderprogramme: Der Leistungsumfang des Cloud Computings nimmt stetig zu.[4] Das *Bundesamt für Sicherheit in der Informationstechnik* fasst diese Entwicklung treffend zusammen. Cloud Computing ist nicht länger ein „Hype [..], sondern […] ist in vielen Bereichen bereits zum Standard avanciert."[5]

Die vorliegende Projektarbeit befasst sich mit dem Thema Private Cloud Computing. Aus der Vielzahl verschiedener Systeme auf dem Markt werden einige ausgewählt und verglichen. Die Kriterien hierfür sind zum einen das Ablaufen auf unternehmenseigenen Servern und zum anderen die Möglichkeit der Einsicht in den Quellcode. Für *.Versicherung* werden diese vorausgewählten Cloud Lösungen mit Hilfe eines zuvor erarbeiteten Kriterienkatalogs bewertet und verglichen. Es werden Scorecards angefertigt, anhand derer das Autorenteam eine Implementierungsempfehlung für das Unternehmen ableitet.

1.2 Zielsetzung

Bei der Erstellung dieser Projektarbeit verfolgt das Autorenteam das Ziel, eine Bewertungsgrundlage für den Vergleich von Private Cloud Lösungen zu erstellen. Das zu erstellende Bewertungssystem soll in funktionale Kriterien, nicht-funktionale Kriterien und Herstellerkrite-

[1] Barton, T. (2012)
[2] Vgl. Matros, R. (2012), S. 34
[3] Vgl. Rajaraman, V. (2014), S. 242 f.
[4] Vgl. Baun, C. u.a. (2011), S. 1 f.
[5] Vgl. Bundesamt für Sicherheit in der Informationstechnik (o. J.)

rien strukturiert sein und es soll durch eine Variation in der Gewichtungen der Kriterien auf verschiedene Unternehmenssituationen anpassbar sein.

Die wichtigsten Private Cloud Lösungen auf dem Markt sollen anhand der Kriterien objektiv bewertet und miteinander verglichen werden. Das Autorenteam möchte eine konkrete Empfehlung zur Einführung eines Private Cloud Systems bei der *.Versicherung* ableiten.

1.3 Wertbeiträge

Der erste Wertbeitrag des Autorenteams ist die Erarbeitung der wissenschaftlichen Grundlagen zu den Themen Cloud Computing, Open Source Software und Bewertungssysteme.

Der zweite Wertbeitrag des Autorenteams besteht in der Ausarbeitung eines Kriterienkatalogs zur Bewertung von Private Cloud Systemen. Dabei werden mögliche Kriterien aus bereits bestehenden Kriterienkatalogen gesammelt und deren Relevanz gemeinsam mit dem Auftraggeber des Projekts diskutiert. Eine Aufteilung in Ausschlusskriterien und Gewichtungskriterien wird erarbeitet.

Der dritte Wertbeitrag des Autorenteams ist die Auswahl der Private Cloud Lösungen, die im Rahmen dieser Projektarbeit verglichen werden, und deren ausführliche Beschreibung.

Der vierte Wertbeitrag des Autorenteams ist der Vergleich der ausgewählten Private Cloud Systeme mittels Scorecards und die Ableitung einer Empfehlung für die unternehmensspezifische Situation bei der *.Versicherung.*

1.4 Struktur der Arbeit

Diese Projektarbeit gliedert sich in acht Teile:

1) In der **Einleitung** werden die Problemstellung, die Zielsetzung und die Wertbeiträge des Autorenteams dargestellt.
2) Im Abschnitt **Cloud Computing** werden die verschiedenen Cloud Kategorien und Service Modelle vorgestellt und Aspekte zu Datenschutz und Datensicherheit erläutert.
3) Im Abschnitt **Open Source** wird das Konzept der quellcode-offenen Software beschrieben, verschiedene Lizenzmodelle verglichen und eine Abgrenzung zu proprietärer Software vorgenommen.
4) Im Abschnitt **Bewertungssysteme** wird die methodische Vorgehensweise zum systematischen Vergleich der Private Cloud Lösungen dargestellt.

5) Im Abschnitt **Erarbeitung eines Kriterienkatalogs** werden Ausschluss- und Gewichtungskriterien zur Bewertung der Private Cloud Lösungen erarbeitet.

6) Im Abschnitt **Vorstellung der zu vergleichenden Cloud Lösungen** werden acht quellcode-offene Private Cloud Lösungen im Hinblick auf die zuvor genannten Vergleichskriterien vorgestellt.

7) Im Abschnitt **Erstellung der Scorecards** werden in zwei Phasen Scorecards erstellt, mit denen die Private Cloud Lösungen bewertet und verglichen werden.

8) In **Fazit und Ausblick** wird eine Implementierungsempfehlung einer Private Cloud Lösung für die *.Versicherung* ausgesprochen.

2 Cloud Computing

Auch wenn es keine einheitliche Definition für Cloud Computing gibt, so sind die grundlegenden Konzepte und Ziele ähnlich. Es handelt sich um die Bereitstellung und Nutzung von IT-Infrastruktur, die den Zugriff auf Rechnerressourcen, wie beispielsweise Server, Speichersysteme und Anwendungen ermöglichen. Durch die Virtualisierung der Cloud-Dienste kann jeder Nutzer von überall auf die Dienste zugreifen.[6]

Eine weitgehend akzeptierte Definition von Cloud Computing liefert das Bundesamt für Sicherheit in der Informationstechnik:

„Cloud Computing bezeichnet das dynamisch an den Bedarf angepasste Anbieten, Nutzen und Abrechnen von IT-Dienstleistungen über ein Netz. Angebot und Nutzung dieser Dienstleistungen erfolgen dabei ausschließlich über definierte technische Schnittstellen und Protokolle. Die Spannbreite der im Rahmen von Cloud Computing angebotenen Dienstleistungen umfasst das komplette Spektrum der Informationstechnik und beinhaltet unter anderem Infrastruktur (z. B. Rechenleistung, Speicherplatz), Plattformen und Software."[7]

Die folgenden Abschnitte beschreiben die fünf Charakteristiken des Cloud Computings, drei Servicemodelle und vier verschiedene Einsatzmodelle, die das National Institut of Standards and Technology (NIST) definiert. [8] Anschließend werden verschiedene Aspekte zum Datenschutz und zur Datensicherheit erläutert.

[6] Vgl. Baun, C.,u.a. (2011), S.4f.
[7] Bundesamt für Sicherheit in der Informationstechnik (o.J.b)
[8] Mell, P./ Grance, T. (2011)

2.1 Charakteristiken von Cloud Systemen

Nach dem NIST gibt es fünf kennzeichnende Charakteristiken für Cloud Lösungen:[9,10]

On-Demand Self-Service – „Selbstbedienung nach Bedarf"

Die Cloud Dienste können je nach Bedarf, automatisch und ohne Interaktion mit dem Anbieter genutzt werden. Der Nutzer kann beispielsweise die Speicherkapazität selbstständig hinzu- oder abbuchen.

Broad Network Access- „Breitgefächerter Zugriff über das Netzwerk"

Die Dienste sind nicht an einen Client gebunden, sondern werden über das Netz bereitgestellt. Der Nutzer kann aus einer Vielzahl von Geräten für den Zugriff auswählen, wie beispielsweise Notebooks, Tablets und Smartphones.

Resource Pooling – „Bündelung der Ressourcen"

Mehrere Anwender bedienen sich aus einem standortunabhängigen Pool an Ressourcen. Diese Art der Ressourcenverteilung wird *Multi-Tenant Modell* genannt. Der Nutzer hat zunächst keinen Einfluss darauf, in welchem Rechenzentrum seine angefragten Dienste bearbeitet werden. Jedoch können die teilnehmenden Nutzer den möglichen Standort des Rechenzentrums vertraglich einschränken.

Rapid Elasticity – „Schnelle Elastizität"

Die Nutzung der Dienste ist auf die Nachfrage angepasst, d.h. die Cloud Dienste werden flexibel und elastisch bezogen oder zurückgegeben. Für den Kunden erscheinen die Ressourcen unendlich.

Measured Service – „Messbarkeit der Nutzung"

Cloud Systeme kontrollieren und optimieren die Ressourcennutzung und sorgen für eine höhere Transparenz sowohl bei den Cloud Anbietern als auch bei den Nutzern. Die Messungen werden dabei in einer an die Art der Ressource angepassten Form durchgeführt, wie beispielsweise die Speicherkapazität bei der Verwendung von Festplattenspeichern.

2.2 Servicemodelle

[9] Mell, P./ Grance, T. (2011)
[10] Busch, D. (2012)

Es existieren drei verschiedene Servicemodelle: Infrastructure as a Service, Platform as a Service und Software as a Service. Abbildung 1 visualisiert die im Folgenden vorgestellten Servicemodelle.

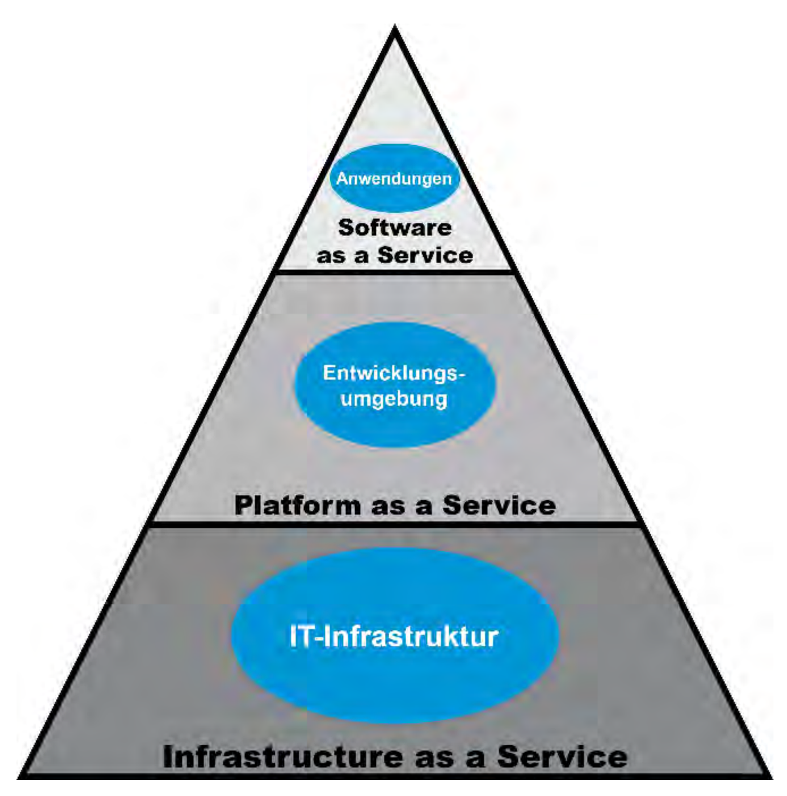


Abbildung 1: Servicemodell-Pyramide

2.2.1 Infrastructure as a Service – Iaas

Infrastructure as a Service bezeichnet im Cloud Computing die webbasierte Bereitstellung von IT-Infrastruktur. Es werden IT-Ressourcen wie z. B. Rechenleistung, Datenspeicher und Netzwerke als Dienst virtuell angeboten. Die Infrastruktur-Nutzung ist an den Bedarf des Nutzers anpassbar. Der Nutzer hat die Kontrolle über einige Komponenten der Infrastruktur, wie beispielsweise das Betriebssystem, den Speicher und die benutzen Anwendungen. Der Großteil der Infrastruktur wird jedoch vom Cloud-Anbieter kontrolliert.[11]

2.2.2 Platform as a Service – PaaS

Platform as a Service bezeichnet die virtuelle Bereitstellung von Laufzeit- und Entwicklungsumgebungen. Der Cloud Provider hat die Kontrolle über die zugrunde liegende Infrastruktur.[12] Die Abrechnung erfolgt nach der Nutzung der Cloud Dienstleistungen.[13]

2.2.3 Software as a Service – SaaS

Software as a Service bezeichnet sämtliche Anwendungen, die den Charakteristiken des Cloud Computings entsprechen. Sowohl die Software als auch die IT-Infrastruktur werden

[11] Vgl. Mell, P./ Grance, T. (2011)
[12] Vgl. Meinel, C. u.s. (2001), S. 32f.
[13] Vgl. Microsoft (2015)

vom Cloud Anbieter zur Verfügung gestellt und können vom Kunden unmittelbar genutzt werden. Durch Konfigurationseinstellungen können die Anwendungen bis zu einem bestimmten Grad personalisiert werden.[14]

2.3 Einsatzmodelle

Es existieren vier Einsatzmodelle: Private Cloud, Public Cloud, Hybrid Cloud und Community Cloud. Abbildung 2 stellt den Zusammenhang der im Folgenden vorgestellten Cloud Einsatzmodelle dar.

Abbildung 2: Einsatzmodelle des Cloud Computings

2.3.1 Private Cloud

In der Private Cloud hat nur ein Unternehmen oder eine Organisation Zugriff auf die Cloud Infrastruktur. Vorteil dieses Cloud Modells ist vor allem der Sicherheitsaspekt: Die Kontrolle interner Daten bleibt innerhalb des Unternehmens. Das Management der Cloud ist entweder unternehmensintern oder wird extern vom Cloud-Anbieter übernommen.[15]

2.3.2 Public Cloud

In der Public Cloud werden die IT-Dienstleistungen öffentlich angeboten. Die Dienstleistungen der Public Cloud sind für eine beliebige Zahl von Privatpersonen und Unternehmen über ein Web-Portal zugänglich. Für den Cloud Anbieter ist dies die kostengünstigste Variante, da

[14] Vgl. Meinel, C. u.s. (2001), S. 34
[15] Vgl. Vollmer, T. (2013), S.25

er nur eine Infrastruktur kontrollieren muss. Meist erhält der Anwender keine Kenntnis darüber, an welchem Ort bzw. in welchem Rechenzentrum die Daten gespeichert werden.[16]

2.3.3 Hybrid Cloud

In der Hybrid Cloud wird die Public Cloud um Aspekte der Private Cloud ergänzt. Der Sicherheitsaspekt der Private Cloud wird mit der Flexibilität und Skalierbarkeit der Public Cloud kombiniert. Der Anwender entscheidet selbst, welche Informationen in der Private oder in der Public Cloud gespeichert werden.[17]

2.3.4 Community Cloud

In der Community Cloud schließen sich mehrere Unternehmen der gleichen Branche zusammen und verbinden ihre Private Cloud Systeme. Diese Möglichkeit wird vor allem zur gemeinsamen Nutzung von Daten und Anwendungen eingesetzt.[18]

2.4 Datenschutz und Datensicherheit

Ein sehr wichtiger und immer wieder umstrittener Aspekt des Cloud Computings ist der Datenschutz. Hier gibt es einige Sicherheits- und Datenschutzanforderungen, die in nationalen und internationalen Gesetzen und Vorgaben festgehalten sind. In Deutschland gilt neben der europäischen Datenschutzrichtlinie 95/46/EG das Bundesdatenschutzgesetz (BDSG). Im internationalen Vergleich variiert das Niveau des Datenschutzes durch verschiedene nationale Gesetzesbungen stark. Es findet jeweils die Gesetzesgrundlage des Landes Anwendung, in dem das Rechenzentrum zur Datenspeicherung steht. Für diese Projektarbeit beschränkt sich das Autorenteam auf die Darstellung der in Deutschland relevanten Gesetze zum Datenschutz. Unternehmen müssen zudem Compliance-Anforderungen (in Deutschland sowie in anderen Ländern) erfüllen, wenn sie ihre Daten in eine Cloud auslagern.[19]

Mit Datenschutz ist der gesetzliche Schutz von personenbezogenen Daten gemeint. Hier findet das deutsche Datenschutzrecht Anwendung.[20] Personenbezogene Daten werden gem. § 3 Abs. 1 des BDSGs als „Einzelangaben über persönliche oder sachliche Verhältnisse einer bestimmten oder bestimmbaren natürlichen Person (Betroffener)“ definiert.

Grundsätzlich bedarf es nach §4 BDSG einer Einwilligung der betroffenen Person oder einer speziellen Erlaubnis durch einen Gesetzestext, um seine Daten zu verarbeiten. Die Daten dürfen nach Einwilligung nur in einem festgelegten Rahmen bearbeitet werden. Public Cloud Anbieter stoßen bei dieser Vorgabe auf Probleme, da weder Standort noch Umfang der Datenverarbeitung bekannt sind. Transparenter ist die Private Cloud, bei der in gewissem Maße

[16] Vgl. Szer, B. (2013), S.61
[17] Vgl. Vollmer, T. (2013), S.26
[18] Vgl. IT Wissen (2015)
[19] Vgl. Fallenbeck, N./ Windorst, I. (2014)
[20] Vgl. Erhardt, S. (o.J.)

der Standort und der Umfang festgelegt werden. Grundsätzlich ist ein hoher technischer und organisatorischer Aufwand notwendig, um eine Public Cloud gemäß dem Datenschutzgesetz aufzubauen. Deshalb ist sie für die Verarbeitung von personenbezogenen Daten eher ungeeignet.[21]

Eine datenschutzrechtliche Auftragsdatenverarbeitung liegt vor, wenn mehr als ein Unternehmen mit den Daten in der Cloud arbeiten. Eine Auftragsdatenverarbeitung ist nach § 11 BDSG eine Weitergabe von personenbezogenen Daten des Auftraggebers an den Auftragnehmer, der diese Daten dann weiterverarbeitet. Der Auftraggeber, also der Cloud Anwender, ist verantwortlich für die Einhaltung der Vorschriften des BDSGs. § 11 BDSG findet nur für eine Datenverarbeitungsübertragung im technischen Sinne Anwendung und ist keine Rechtsgrundlage für eine inhaltliche Aufgabenübertragung.[22]

Zusätzlich zu den gesetzlichen Vorschriften sind auch Zertifizierungen und Gütesiegel der Cloud Anbieter als Nachweis zur Einhaltung von Sicherheits- und Datenschutzanforderungen zu beachten. Beispiele hierfür sind:

- EuroCloud SaaS Star Audit
- ISO 27001
- das Europäische Datenschutz-Gütesiegel EuroPriSe
- TÜV Trusted Cloud Zertifikat
- Federal Risk and Authorization Program (FedRAMP)

Die verschiedenen Zertifikate unterscheiden sich in ihren Sicherheitsanforderungen. Grundsätzlich sind neben den Zertifikaten die Allgemeinen Geschäftsbedingungen des Cloud Anbieters zu beachten. Andere wichtige Aspekte werden vertraglich geregelt.[23]

Weitere Maßnahmen zum Schutz von personenbezogenen Daten sind die Anonymisierung und die Pseudonymisierung durch eine Software. Bei der Anonymisierung wird ein Teil der personenbezogenen Daten an einem anderen Speicherort abgelegt. Bei der Pseudonymisierung werden die personenbezogenen Daten durch ein Pseudonym, beispielsweise einer Zahlenkombination, ersetzt. Grenzen finden diese Ideen in der praktischen Anwendung, da die Daten bei jeder Abfrage oder Bearbeitung wieder entschlüsselt werden müssen. Zudem bietet der Einsatz des Verschlüsselungsprogramms eine Möglichkeit für Hackerangriffe.[24]

Abzugrenzen vom Datenschutz ist die **Datensicherheit**, die sich mit der Sicherheitstechnik in Bezug auf Hardware und Software auseinandersetzt. Ein Zeichen der Datensicherheit ist

[21] Vgl. Bitcom (2010), S.62
[22] Vgl. Bitcom (2010), S.64f.
[23] Vgl. Fallenbeck, N./ Windorst, I. (2014)
[24] Vgl. Bitkom (2010), S.69f.

die Aktualität der Sicherheitstechnik.[25] Zudem wird die Datensicherheit durch die *Datensicherung* unterstützt. Die Datensicherung ist die redundante Datenspeicherung und stellt somit eine Datensicherungsmaßnahme dar.[26] Im Gesetz des Bundesamts für Sicherheit in der Informationstechnik wird in § 2 Abs. 2 die „Sicherheit in der Informationstechnik“ als die „Einhaltung bestimmter Sicherheitsstandards, die die Verfügbarkeit, Unversehrtheit oder Vertraulichkeit von Informationen betreffen“ definiert. Nach § 9a BDSG können Anbieter Ihre Datenschutz- und Datensicherheitskonzepte durch unabhängige Prüfer testen lassen und ihre Ergebnisse des Tests veröffentlichen. Schon in der Anfangsphase der Softwareentwicklung sollte eine Analyse zur Datensicherheit durchgeführt werden. Es sollten mögliche Szenarien getestet werden, um konkrete Sicherheitsanforderungen zu formulieren und den weiteren Aufbau des Cloud Systems darauf zu basieren.[27]

3 Open Source Software

Unter Open Source Software versteht man Programme, deren Quellcode für Anwender einsehbar und veränderbar ist. In der Regel ist diese Zugänglichmachung für den Nutzer kostenfrei.[28] Sowohl die Entwicklung als auch die Wartung und Verbesserung der Software erfolgt meistens durch mehrere Einzelpersonen oder unabhängige Organisationen.[29] Open Source Software kann in verschiedenen Formen auftreten, bspw. in eingegrenzten Programmteilen, in eigenständigen Programmen oder in umfassenden Programmbibliotheken.[30]

§69c des Urheberrechtsgesetzbuches (UrhG) listet Handlungen im Zusammenhang mit Computerprogrammen auf, die der Zustimmung des Urhebers bedürfen.[31] Diese sind Vervielfältigung (§69c Nr. 1), Bearbeitung (§69c Nr. 2), Verbreitung (§69c Nr. 3) und öffentliche Wiedergabe (§69c Nr. 4). Allgemein lässt sich sagen, dass die Entwicklergruppe von Open Source Software seinen Nutzern diese Handlungen einräumt. Insbesondere für die Bearbeitung wird der Quellcode der Open Source Software von seinen Entwicklern offengelegt.[32] Die auf dem Markt verfügbaren Open Source Lizenzen unterscheiden sich in den Details dieser Nutzungsrechte.

[25] Vgl. a7digital GmbH (o.J.)
[26] Vgl. Bedner, M. (2012), S.158f.
[27] Vgl. Lissen, N./Brünger, C./ Damhorst, S (2014) S.30f
[28] Vgl. Keßler, S. (2013), S. 1
[29] Vgl. Open Source Initiative (o. J.)
[30] Vgl. Haun, M. (2014), S. 417
[31] Vgl. §69c UrhG
[32] Vgl. Hüttenegger, G. (2006), S. 4

3.1 Open Source Lizenzen

Im Folgenden werden die beiden für diese Projektarbeit relevanten Open Source Lizenzen vorgestellt: die General Public License und die Apache Lizenz.

3.1.1 General Public License

Die General Public License wurde von der Free Software Foundation erarbeitet. Im Juni 2007 wurde die aktuelle Version 3.0 veröffentlicht. Software unter der General Public License darf vom Nutzer vervielfältigt, bearbeitet, verbreitet und öffentlich zugänglich gemacht werden. Die Lizenzvereinbarung unterstreicht, dass Support-Aktivitäten und Garantieleistungen auch entgeltlich verrichtet werden dürfen.

Personen, die General Public lizenzierte Software modifizieren, müssen Änderungen an der Software vor ihrer weiteren Distribution kenntlich machen und rechtliche Hinweise auf jeder Benutzerschnittstelle platzieren. Die wichtigste Forderung der General Public License besteht darin, dass jede Software, die Teile eines General Public lizenzierten Programms enthält oder aus einem solchen heraus entwickelt wurde, ebenfalls unter die Bedingungen der General Public License gestellt wird.[33]

3.1.2 Apache Lizenz

Die Apache Lizenz wird von der Apache Software Foundation verwaltet. Die aktuelle Version 2.0 wurde im Januar 2004 veröffentlicht. Ein Computerprogramm, das unter der Apache Lizenz steht, räumt seinen Nutzern eine allgemeingültige, kostenfreie und unwiderrufliche Lizenz zur Vervielfältigung, Bearbeitung, Verbreitung, öffentlichen Wiedergabe und Sublizenzierung der Software ein.

Der Nutzer verpflichtet sich dabei zur Einhaltung der folgenden Bedingungen:

- Jeder Nutzer der Originalsoftware oder einer modifizierten Version des Programms erhält eine Kopie der Apache Lizenzvereinbarung.
- Änderungen an der Software müssen dem Nutzer eindeutig kenntlich gemacht werden.
- Die Urheberrechtsbestimmungen der Originalsoftware müssen in den Quellcode einer veränderten Software aufgenommen werden.

[33] Vgl. Free Software Foundation (2007)

Im Gegensatz zu Software, die unter der General Public License steht, müssen Computerprogramme, die Apache lizenzierten Quellcode enthalten, nicht ebenfalls unter die Apache Lizenz gestellt werden.[34]

3.2 Abgrenzung zu proprietärer Software

Im Zusammenhang mit Open Source Software werden häufig auch die Begriffe proprietäre Software, kommerzielle Software, Public Domain Software, Shareware, Freeware und Copylefted Software genannt. Die Literatur bildet keine einheitlichen Definitionen dieser Begriffe ab. Im nachfolgenden Abschnitt werden deshalb die für diese Projektarbeit gültigen Definitionen erarbeitet und in den Kontext der Open Source Software eingeordnet.

Proprietäre Software räumt seinen Nutzern zunächst keine der nach §69c UrhG zustimmungsbedürftigen Handlungen ein. Der Nutzer darf die proprietäre Software also ohne die vorherige Zustimmung der Entwickler weder vervielfältigen, noch bearbeiten oder verbreiten und öffentlich zugänglich machen. Das wichtigste Unterscheidungsmerkmal zwischen proprietärer Software und Open Source Software ist, dass dem Nutzer der Zugang zu einer nutzbaren Form des Quellcodes der proprietären Software verweigert wird.[35]

Kommerzielle Software räumt seinen Nutzern keine Vervielfältigung, Bearbeitung, Verbreitung oder öffentliche Zugänglichmachung ein. Für die Benutzung der kommerziellen Software erheben die Entwickler ein Entgelt.[36]

Shareware räumt seinen Nutzern ein, die Software in einem zeitlich begrenzten Zeitraum zu testen. In diesem Zeitraum darf die Software vervielfältigt werden und weiteren Interessenten angeboten werden. Für eine Nutzung über Testzwecke hinaus verlangen die Entwickler der Shareware ein Entgelt. In den meisten Fällen wird der Quellcode der Shareware nicht offengelegt, sodass der Nutzer keine Modifikationen vornehmen kann.[37]

Freeware räumt seinen Nutzern ein, die Software zu vervielfältigen, zu verbreiten und öffentlich zugänglich zu machen. Eine Modifikation des Quellcodes wird den Nutzern jedoch nicht gestattet. Der Nutzer erhält keinen Zugang zum Quellcode der Software.[38]

[34] Vgl. Apache Software Foundation (2004)
[35] Vgl. Hunziker, S. / Rihs, S. (2006), S. 332
[36] Vgl. Picot, A. / Fiedler, M. (2008), S. 166
[37] Vgl. Free Software Foundation (2014)
[38] Vgl. Ruckdeschel, R. (o. J.)

Public Domain Software räumt seinen Nutzern uneingeschränkt alle zustimmungsbedürftigen Handlungen nach §69c UrhG ein. Zumeist wird dem Nutzer der Quellcode der Public Domain Software offengelegt.[39]

Copylefted Software räumt seinen Nutzern ein, die Software zu vervielfältigen, zu bearbeiten, zu verbreiten und öffentlich zugänglich zu machen. Der Quellcode der Copylefted Software wird dem Nutzer offengelegt. Der Nutzer verpflichtet sich den Quellcode jeder Modifikation und jeder Weiterentwicklung der Software offen zu legen und ebenfalls unter eine Copylefted Softwarelizenz zu stellen.[40]

Abbildung 3 veranschaulicht den Zusammenhang der verschiedenen Softwaredistributionsformen.

Abbildung 3: Softwaredistributionsformen

[39] Vgl. Institut für Rechtsfragen der Freien und Open Source Software (o. J.)
[40] Vgl. Sury, U. (2007), S. 45

4 Bewertungssysteme

Für die Bewertung der einzelnen Cloud Lösungen wird die Methode der Scorecards des Projektmanagements angewendet. Scorecards haben den Vorteil, dass sie numerische und nicht-numerische Entscheidungskriterien gleichzeitig abbilden. Der Vergleich der verschiedenen Cloud Lösungen erfolgt basierend auf einem umfassenden Kriterienkatalog.[41]

Das Projektteam führt die Bewertung in zwei Phasen durch:

- In der ersten Phase wird das **Unweighted 0-1 Factor Model** angewendet. In dieser Scorecard werden Ausschlusskriterien, wie bspw. eine aktive Community, geprüft. Die Cloud Lösungen, die diese Ausschlusskriterien nicht aufweisen, werden aussortiert und in der zweiten Bewertungsphase nicht berücksichtigt.
- In der zweiten Phase wird das **Weighted Factor Scoring Model** angewendet. Neben Ausschlusskriterien enthält die Scorecard hierbei auch Gewichtungskriterien. Durch die Bewertung der zweiten Phase enthält jede Cloud Lösung eine individuelle Punktzahl, anhand der die Cloud Lösungen in eine Reihenfolge gebracht werden.

4.1 Unweighted 0-1 Factor Model

Für das *Unweighted 0-1 Factor Model* wird zunächst eine Liste erstellt, die alle relevanten Ausschlusskriterien enthält. Im Anschluss wird für alle Lösungsansätze geprüft, ob diese ein Kriterium erfüllen oder nicht. Ein erfülltes Kriterium wird mit *1* bewertet; ein nicht erfülltes Kriterium mit *0*. Die Bewertung kann dabei von den Team-Mitgliedern getrennt voneinander oder in einer Gruppendiskussion ausgefüllt werden. Für die Durchführung dieses Projektes entscheidet sich das Autorenteam dazu, eine Konsensmeinung in einer Gruppendiskussion zu erreichen. Nachdem die Bewertungen abgegeben wurden, wird für jeden Lösungsansatz eine Punktzahl ermittelt. Diejenigen Lösungen, die eine bestimmte Punktzahl nicht erreicht haben, erfüllen die Ausschlusskriterien nur unzureichend und scheiden aus.[42]

Tabelle 1 stellt ein Beispiel einer Scorecard des *Unweighted 0-1 Factor Models* dar.

	Cloud A	**Cloud B**	**Cloud C**
Kriterium 1	1	0	1
Kriterium 2	1	1	0

[41] Vgl. Winter, S. (2014), S. 143 f.
[42] Vgl. Meredith, J./Mantel, S. (2009), S. 51

Kriterium 3	1	0	1
Summe	**3**	**1**	**2**

Tabelle 1: Beispiel einer Scorcard des Unweighted 0-1 Factor Model

4.2 Weighted Factor Scoring Model

Das *Weighted Factor Scoring Model* übernimmt die zuvor erstellte Liste aus Ausschlusskriterien und ergänzt diese um Gewichtungskriterien. Diese Kategorisierung ist analog zu einer Einteilung in Muss- und Kann-Kriterien.[43] Die einzelnen Kriterien der Liste werden anhand ihrer Wichtigkeit in eine Reihenfolge gebracht und mit Prozentpunkten relativ zueinander gewichtet. Die Bewertung der Kriterien für die verschiedenen Cloud Lösungen erfolgt – wie auch im Unweighted 0-1 Factor Model – in unabhängigen Bewertungen der Teammitglieder oder in einer Teamdiskussion. Das Autorenteam entscheidet sich wieder für die Durchführung einer Gruppendiskussion. Anstelle der 0-1-Bewertung steht in der zweiten Bewertungsrunde eine numerische Skala von 1-5, die Abstufungen zwischen den Lösungsansätzen ermöglicht. Durch Multiplikation der abgegebenen Bewertungen mit der zugeordneten Prozentzahl wird eine individuelle Punktzahl für jedes Cloud System ermittelt. Die Lösung mit der höchsten Punktzahl ist für die Unternehmenssituation die geeignetste.[44]

Tabelle 2 stellt ein Beispiel einer Scorecard des W*eighted Factor Scoring Models* dar.

		Cloud A	Cloud C
Kriterium 1	40%	5	3
Kriterium 2	35%	4	1
Kriterium 3	20%	2	4
Kriterium 4	5%	4	4
Summe	100%	**4**	**2,55**

Tabelle 2: Beispiel einer Scorecard des Weighted Factor Scoring Models

5 Erarbeitung eines Kriterienkatalogs

Um Cloud Lösungen sinnvoll zu bewerten ist es notwendig, zuerst einen Kriterienkatalog zu erstellen und Ausschlusskriterien sowie gewichtende Kriterien zu definieren. Mithilfe bereits existierender Kriterienkataloge werden einige Kriterien erarbeitet und dem Auftraggeber die-

[43] Vgl. Hettl, M. (2013), S. 195
[44] Vgl. Meredith, J./Mantel, S. (2009), S. 53 ff.

ser Projektarbeit vorgestellt. Die Bewertungskriterien aus dem Vergleich von zehn Private Cloud Lösungen der Zeitschrift „Computerwoche“[45] geben erste Anhaltspunkte für den Kriterienkatalog dieser Projektarbeit.

Das Autorenteam zieht weitere Dokumente heran, um ein umfassendes Verständnis über die vielfältigen Bewertungskriterien von Cloud Systemen zu bekommen. Darunter befinden sich:

- das Eckpunktepapier „Sicherheitsempfehlungen für Cloud Computing Anbieter“ des Bundesamtes für Sicherheit in der Informationstechnik[46]
- der Leitfaden „Cloud Computing - Was Entscheider wissen müssen“ des Bundesverbandes Informationswirtschaft, Telekommunikation und neue Medien e.V. (BITKOM)[47] und
- das „SLA Richtliniendokument für Cloud Dienstleistungen“ des Projektes Autonomic SLA Management as a Service (ASLAMaaS)[48].

Aus diesen Quellen erstellt das Autorenteam einen ersten, eigenen Kriterienkatalog.

Im Gespräch mit dem Auftraggeber dieser Projektarbeit wird der erste Entwurf des Kriterienkataloges diskutiert und die Kriterien definiert, welche von besonderer Bedeutung für die unternehmensspezifische Situation der *.Versicherung* sind. Mithilfe dieser Erkenntnisse sowie den bereits existierenden Kriterienkatalogen zur Bewertung von Cloud Systemen wird der im Folgenden vorgestellte Kriterienkatalog erarbeitet. Dabei wird zwischen funktionalen Kriterien, nicht-funktionalen Kriterien und Herstellerkriterien unterschieden.

5.1 Funktionale Kriterien

Funktionale Kriterien beziehen sich auf die technische Funktionsweise der Lösung. Dazu zählen auch die Servicemodelle, die in Kapitel 2.2 „Servicemodelle“ beschrieben werden. Folgende Kriterien werden als funktionale Kriterien identifiziert:

5.1.1 Kontroll- und Analysemechanismen

Für das Management von Cloud Systemen im Unternehmen ist es wichtig, über die aktuelle Auslastung des Systems informiert zu sein und somit mögliche Engpässe vorauszusehen und ihnen entgegenzuwirken. Gerade beim Einsatz von Cloud Systemen in Unternehmen herrscht eine große Abhängigkeit vom Cloud Anbieter. Somit wird eine gute Überwachung

[45] Vgl. Hackmann, J. (2014), S.1ff
[46] Vgl. Essoh D./ Doubrava, Dr.C./ Münch, I. (2012), S.1ff
[47] Vgl. Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e.V. (2010), S.1ff
[48] Vgl. Frey, S. (2014), S.1ff

des Cloud Systems notwendig, um das Cloud System autark verwalten und steuern zu können. Auch um die unternehmensinterne Abrechnung der Kosten vorzunehmen, wird die Nutzungsdauer und -intensität der verschiedenen Abteilungen gemessen.

Wichtig bei einem sogenannten „Monitoring“ ist, dass die komplette Cloud Lösung immer überwacht und analysiert wird. Cloud Lösungen sollten Mechanismen zur Analyse der Auslastung (z.B. CPU- und Netzauslastung), Nutzerentwicklung und zum Datenverkehr (z.B. durchschnittliche Transaktionszeit) bereitstellen, um ein optimales Cloud Management zu ermöglichen.[49] Dabei müssen die Zeiträume der Ressourcenmessungen (Intervallzeit) und der Reportgenerierung (Reportintervall) minimal sein.[50]

5.1.2 Betriebssystemkompatibilität

Cloud Systeme können mit unterschiedlichen Betriebssystemen verwendet werden. Manche Cloud Lösungen stellen zusätzlich eine Applikation bereit, die einen Zugriff von mobilen Endgeräten aus ermöglicht. Durch eine höhere Anzahl kompatibler Betriebssysteme erhöht sich die Flexibilität für ein Unternehmen, die Cloud Lösung einzusetzen. Für die Bewertung der Cloud Lösung wird die Anzahl der kompatiblen Betriebssysteme als Kennzahl herangezogen.

5.1.3 Virtualisierungstechnologie

Im Cloud Computing werden sogenannte „Hypervisor“ verwendet. Ein Hypervisor ist eine Virtualisierungsschicht,[51] die es ermöglicht, eine virtuelle Umgebung zu erschaffen und letztendlich die virtuellen Maschinen auf den Servern zu generieren.[52]

Die Verwendung von virtuellen Maschinen erlaubt verschiedenen Nutzern, zeitgleich auf die bereitgestellten Dienste aus der Cloud zuzugreifen. So ist eine Migration von physischen zu virtuellen, genauso wie von virtuellen zu physischen und zwischen virtuellen Maschinen möglich.[53] Das Autorenteam entscheidet sich dazu, die Anzahl der verschiedenen Virtualisierungsmöglichkeiten, sprich Hypervisor, einer Cloud Lösung als Kennzahl zur Bewertung der Virtualisierungstechnologie zu verwenden, da diese auf eine flexible Einsetzbarkeit der Cloud Lösung hinweisen.

[49] Vgl. Essoh D./ Doubrava, Dr.C./ Münch, I. (2012), S. 47
[50] Vgl. Frey, S. u.a. (2014), S.31
[51] Vgl. Rouse, M. (2013)
[52] Vgl. Md.Mahbub-E-Noor (2014), S.1ff
[53] Vgl. Steven, J. (2014)

5.2 Nicht-funktionale Kriterien

Nicht-Funktionalen Kriterien beziehen sich auf die Aspekte der Cloud Lösung, die nicht in direktem Zusammenhang zur Funktionsweise des Produktes stehen. Folgende Kriterien wurden als nicht-funktionale Kriterien identifiziert:

5.2.1 Management der Cloud

Für den Einsatz von Private Cloud Systemen im Unternehmen ist es essentiell wichtig, Nutzer, Ressourcen und Services zu verwalten. Eine Möglichkeit dazu bietet ein Self Service Portal.

5.2.1.1 Self Service Portal

Ein Self Service Portal ist eine Plattform, in welcher dedizierte, authentifizierte Nutzer Dienste hinzu- und abbuchen können. Die Nutzer entscheiden selbst, wann sie welche Optionen benötigen und verwalten das Cloud System unternehmensintern, d.h. ohne „Umweg“ über ein Supportteam. Ein Self Service Portal bietet außerdem die Möglichkeit, kundenspezifische Dienstleistungen bereitzustellen. Durch ein Self Service Portal wird die Organisation und Verwaltung der Private Cloud stark vereinfacht.[54]

5.2.1.2 Unternehmensintegration

Die unproblematische Einbindung einer Private Cloud Lösung ist ebenfalls von entscheidender Bedeutung. Bestehende Systeme und Prozesse sollten durch die Cloud Lösung nicht beeinträchtigt, sondern unterstützt werden. Das Management der Cloud Lösung muss sich ohne Mehraufwand in die Verwaltung der übrigen IT-Infrastruktur integrieren lassen.[55]

5.2.2 Skalierbarkeit

Da Unternehmen sich heutzutage sehr schnell dynamisch verändern ist es wichtig, bei der Auswahl einer Private Cloud Lösung auf ihre Skalierbarkeit zu achten. Benutzer und Speicherplatz müssen problemlos hinzu- oder abgebucht werden (Up- oder Downscaling). Skalierbare Cloud Lösungen ermöglichen die schnelle Anpassung der Ressourcen an geschäftliche Anforderungen und bieten dadurch eine hohe Flexibilität. Auch die Kosten entwickeln sich verbrauchsabhängig und können genau gesteuert werden.[56]

[54] Vgl. Hackmann, J. (2014), S.1ff
[55] Vgl. Repschläger, J. (2013), S.33
[56] Vgl. Blaisdell, R. (2012)

5.2.3 Datenschutz und Datensicherheit

Cloud Lösungen weisen verschiedene Mechanismen zum Schutz und zur Sicherung der Daten auf. Dazu zählen u.a. Firewalls, redundant aufgebaute Umgebungen, unterschiedliche Zugriffsreche und eine automatisierte Sicherung der Daten.

5.2.4 Verlässlichkeit

Wenn künftig komplexe Prozesse oder Datenbanken über ein Cloud System abgewickelt werden, ist eine hohe Verlässlichkeit des Systems wichtig. Dazu werden durchschnittliche Ausfallraten und -zeiten sowie Erfahrungen anderer Kunden mit dem Produkt betrachtet. Da im Rahmen dieses Projektes keine langfristige Überprüfung einzelner Cloud Systeme möglich ist, stützt sich die Bewertung der Verlässlichkeit ausschließlich auf Erfahrungswerte bestehender Kunden.

5.2.5 Benutzerfreundlichkeit

Die Bedienbarkeit des Systems sollte intuitiv und einfach sein. Dazu muss ein Self Service Portal (falls vorhanden) benutzerfreundlich gestaltet sein. Die Benutzerfreundlichkeit einer Cloud Lösung ist stark von den individuellen Vorlieben des Benutzers abhängig. Eine Bewertung ist deshalb sehr subjektiv.

5.2.6 Installationsprozess

Der Installationsprozess einer Private Cloud Lösung sollte einfach, schnell und problemlos sein. Bei einigen Produkten kann das Autorenteam selbst eine Demoversion installieren und den Installationsprozess anhand der dabei gemachten Erfahrungen bewerten. Die Lösungen, welche keine Demoversion zur Verfügung stellen, werden anhand der Kommentare und Erfahrungen anderer Nutzer in Communities und Foren bewertet.

5.3 Herstellerkriterien

Herstellerkriterien bezeichnen alle außerhalb des Funktionsumfangs liegenden Aspekte einer Cloud Lösung, die vom Hersteller des Produktes abhängig sind. Folgende Kriterien werden als Herstellerkriterien definiert.

5.3.1 Innovations- und Releasegeschwindigkeit

Im Bereich des Cloud Computings ändern sich Anforderungen, technische Möglichkeiten, Innovationen und Standards von Tag zu Tag. Eine Private Cloud sollte daher regelmäßig verbessert und vom Anbieter geprüft und gepflegt werden. Die Geschwindigkeit, mit der

neue Funktionen und Updates entwickelt und bereitgestellt werden, variiert von Hersteller zu Hersteller. Ein schneller Release Zyklus der letzten Jahre ist von Vorteil und deutet auf ein zukunftsfähiges Produkt hin.[57] Um die Innovations- und Releasegeschwindigkeit der Cloud Lösungen zu vergleichen, wird der Durchschnitt der bisherigen Hauptversionen seit der Veröffentlichung ermittelt und als Kennzahl verwendet.

5.3.2 Support

Für ein im Unternehmen eingesetztes Cloud Produkt ist meistens ein Supportservice des Anbieters etabliert. Probleme, die die Cloud Nutzer nicht selbst lösen können, werden in einer zentralen Anlaufstelle aufgenommen und behandelt. Die Supportleistung wird anhand der Erreichbarkeit und der Qualität des Supports bewertet. Verschiedene Kennzahlen (z.B. durchschnittliche Reaktionszeit und Anzahl gelöster Tickets) können gemessen und in einem Service Level Agreement (SLA) mit dem Anbieter vereinbart werden.[58] Bei vielen Anbietern gibt es verschiedene Servicemodelle zur Auswahl. Bereitgestellte Dokumente und Tutorials werden ebenfalls in die Bewertung des Kriteriums Support aufgenommen. Das jeweils umfassendste Servicemodell und die darin enthaltenen Dienste werden im Vergleich der Lösungen zur Bewertung des Kriteriums verwendet.

5.3.3 Community

Die Nutzer eines Cloud Systems tauschen sich auf Webportalen und -foren mit anderen Anwendern aus und profitieren von deren Erfahrungen.[59] Für Anwender einer Cloud Lösung ist wichtig, wie eine solche Community organisiert ist, ob sie gepflegt und gewartet wird und welche Zugangsrechte herrschen. Qualität, Anzahl der Benutzer und die Reaktionszeit auf gestellte Fragen sind die Grundlage für die Bewertung dieses Kriteriums.

Abbildung 4 bietet einen Überblick über die einzelnen Kriterien des Kriterienkatalogs.

[57] Vgl. Büst, R. (2014a)
[58] Vgl. Repschläger, J. (2013), S.31
[59] Vgl. Büst, R. (2014a)



Abbildung 4: Kriterienkatalog

5.4 Ausschlusskriterien

Ausschlusskriterien sind in Absprache mit dem Auftraggeber der Projektarbeit definiert. Wie in Kapitel 4.1 „Unweighted 0-1 Factor Model“ bereits beschrieben, scheiden Lösungen aus, welche die Ausschlusskriterien nicht erfüllen. Aus den oben vorgestellten Kriterien werden sechs Ausschlusskriterien definiert, die in Absprache mit dem Auftraggeber priorisiert sind. Die Tabelle 3 veranschaulicht die festgelegten Ausschlusskriterien.

Scorecard I
Funktionale Kriterien
Kontroll- und Analysemechanismen
Nicht-funktionale Kriterien
Management der Cloud
Skalierbarkeit

Herstellerkriterien
Innovations- und Releasegeschwindigkeit
Support
Community

Tabelle 3: Ausschlusskriterien

Als Ausschlusskriterien werden in Abstimmung mit dem Auftraggeber der Projektarbeit die Kriterien Kontroll- und Analysemechanismen, Management der Cloud, Skalierbarkeit, Innovations- und Releasegeschwindigkeit, Support sowie Community gewählt.

Kontroll- und Analysemechanismen ermöglichen einen bedarfsgerechten Einsatz der Cloud im Unternehmen und müssen vorhanden sein.

Ebenso ist das **Management der Cloud** ein Ausschlusskriterium. In dieses Kriterium fließen die beiden Unterpunkte Self Service Portal und Unternehmensintegration ein. Nicht **skalierbare** Cloud Lösungen werden mit einer 0 bewertet; skalierbare Cloud Lösungen erhalten die Bewertung 1.

Ein **Releasezyklus** der Hauptversionen der Cloud Lösung von über einem Jahr deutet auf eine inaktive Entwicklergemeinschaft hin und wird vom Autorenteam mit 0 bewertet. Die Bereitstellung einer zentralen Anlaufstellung zur Lösung von auftretenden Problemen wird vom Autorenteam mit 1 bewertet; Lösungen, die keinen **Support** bieten, werden mit 0 bewertet. Das Vorhandensein einer **Community** wird ebenfalls als Ausschlusskriterium definiert. In diesen Punkt fließt nicht die Bereitstellung der Plattform direkt vom Anbieter ein. Für die Bewertung dieses Kriteriums werden IT-Foren nach Beiträgen über die einzelnen Cloud Lösungen untersucht und daran das Vorhandensein einer aktiven Community gemessen. In der zweiten Scorecard wird dieses Kriterium differenzierter betrachtet.

5.5 Gewichtungskriterien

Wie in Kapitel 4.2 „Weighted Factor Scoring Model“ beschrieben, werden alle Kriterien anhand ihrer Wichtigkeit in eine Reihenfolge gebracht und mit Prozentpunkten relativ zueinander gewichtet. Tabelle 4 veranschaulicht die erarbeitete Gewichtung.

Scorecard II	Gewichtung
Funktionale Kriterien	

Kriterium	Gewichtung
Kontroll- und Analysemechanismen	10,00%
Betriebssystemkompatibilität	2,00%
Virtualisierungstechnologie	2,00%
Nicht-funktionale Kriterien	
Management der Cloud: Self Service Portal	5,00%
Management der Cloud: Unternehmensintegration	5,00%
Skalierbarkeit	15,00%
Benutzerfreundlichkeit	5,00%
Datenschutz & Datensicherheit	7,00%
Verlässlichkeit	7,00%
Installationsprozess	2,00%
Herstellerkriterien	
Innovations- und Releasegeschwindigkeit	15,00%
Support	10,00%
Community	15,00%
Gesamtergebnis	**100,00%**

Tabelle 4: Gewichtungskriterien

Die Scorecard II wird in einer Gruppendiskussion vom Autorenteam erstellt und bringt alle zuvor aufgelisteten Kriterien in Relation zueinander. Die Kriterien sind in 5 Stufen gewichtet. Im Vergleich zu den anderen Kriterien eher unwichtige und wenig aussagekräftige Kriterien werden mit 2% bewertet. Die nächste Stufe ist bei 5% für differenziertere Kriterien, gefolgt von 7%. Die höchsten Gewichtungen liegen bei 10% und 15%. Diese Gewichtungen werden für Kriterien vergeben, die in Absprache mit dem Auftraggeber priorisiert sind.

Die Kriterien **Skalierbarkeit, Innovations- und Releasegeschwindigkeit** und die **Community** werden jeweils mit 15% bewertet. Der **Support** und die vorhandenen **Kontroll- und Analysemechanismen** werden mit jeweils 10% ebenfalls sehr hoch bewertet. Das Projektteam testet diese beiden Kriterien anhand der verfügbaren Demoversionen der Cloud Lösungen.

Die nächsthöhere Gewichtung erhält das Kriterium **Datenschutz und Datensicherheit**. Da bei allen vorgestellten Cloud Lösungen die Qualität des Datenschutzes und der Datensicherheit überwiegend hoch und zwischen den Cloud Lösungen kaum zu differenzieren ist, wird dieses Kriterium mit 7% gewichtet. So wird eine abgeschwächte Bewertung durch ein überwiegend positives und sehr stark gewichtetes Kriterium vermieden. Ein ähnliches Phänomen ist bei der Bewertung der **Verlässlichkeit** der Cloud Lösungen vorhanden. Die Verlässlichkeit einer Cloud Lösung kann nur im täglichen Einsatz in einem Unternehmen realitätsgetreu bewertet werden. In dieser Arbeit wird die Verlässlichkeit basierend auf Erfahrungsberichten

anderer Anwender und eigenen Erfahrungen mit der Demoversion bewertet. Es erhält eine Gewichtung von 7%.

Auch die **Unternehmensintegration** kann vom Projektteam nur sehr eingeschränkt getestet und bewertet werden. Für die Bewertung der Unternehmensintegration zieht das Autorenteam Erfahrungsberichte und Fallstudien anderer Nutzer heran. Dieses Kriterium wird mit 5% gewichtet. Auch das Management der Cloud über ein **Self Service Portal** wird mit 5 % gewichtet. Basierend auf dem Umfang der über das Self Service Portal verfügbaren Dienstleistungen werden die Cloud Lösungen bewertet. Das sehr subjektive Kriterium der **Benutzerfreundlichkeit** wird anhand von Demoversionen getestet und bewertet. Benutzerfreundlichkeit wird mit 5% gewichtet.

Mit 2% gewichtet werden der **Installationsprozess**, die **Betriebssystemkompatibilität** und die **Virtualisierungstechnologie** der Cloud Lösungen. Das Autorenteam installiert für einige Cloud Lösungen Demoversionen. Aus den Erfahrungen dieser Installationen kann nur bedingt auf den Installationsprozess der vollständigen Cloud Lösung geschlossen werden. Darum wird der Installationsprozess mit 2% bewertet. Der Mehrwert aus einer hohen Betriebssystemkompatibilität und einer hohen Anzahl von möglichen Virtualisierungstechnologien ist stark von der bestehenden IT-Infrastruktur abhängig. Eine positive Bewertung dieser Kriterien lässt auf eine flexibel einsetzbare Lösung schließen. Die Kriterien Betriebssystemkompatibilität und Virtualisierungstechnologie werden mit 2% bewertet.

6 Vorstellung der zu vergleichenden Cloud Lösungen

Im Folgenden werden die ausgewählten Cloud Lösungen Cloud Foundry, CloudStack, Eucalyptus, Open Nebula, openQRM, OpenStack, ownCloud und Syncthing im Einzelnen vorgestellt.

6.1 Cloud Foundry

Cloud Foundry ist eine Cloud Lösung zum Aufbau eines Platform as a Service Systems.[60] Die Lösung wurde erstmals im Jahr 2011 von der Pivotal Software Incorporation unter der Apache Lizenz veröffentlicht.[61] Im Dezember 2014 wurde die Cloud Foundry Foundation gegründet.[62] Sie wird durch namhafte IT-Unternehmen wie beispielsweise HP, IBM und Intel,

[60] Vgl. Cloud Foundry Foundation (2015a)
[61] Vgl. VMware (2011)
[62] Vgl. Cloud Foundry Foundation (2015b)

unterstützt und strebt für das Jahr 2015 die Etablierung als globalen Industriestandard an.[63] Die Lösung ermöglicht die Verwendung mehrerer Entwicklungsumgebungen und Dienstleistungen im Rahmen einer Cloud Architektur.[64] Cloud Foundry wird als Grundlage, u.a. für die Systeme IBM Bluemix, CenturyLink Cloud und ActiveState genutzt. Die Lösung unterstützt die Programmiersprachen Java, Ruby, Node.js, Scala, Python und PHP. Die Lösung besteht aus mehreren Elementen, die separat heruntergeladen werden:

- Der **Router** leitet eingehenden Traffic an die erforderliche Komponente weiter, meist zum Cloud Controller oder zur auf dem Droplet Execution Agent laufenden Anwendung.
- Der **oAuth2** *Server* und der **Login** *Server* sind für das Identitätsmanagement verantwortlich und ermöglichen eine sichere Authentifizierung der Nutzer.
- Der **Cloud Controller** ist für den Lebenszyklus der Anwendungen verantwortlich und speichert Metadaten der Anwendungen, die anschließend zum Droplet Execution Agent geleitet werden. Außerdem stellt der Cloud Controller Berichte von u.a. Speicherkapazitäten, Serviceinstanzen und Nutzerrollen zur Verfügung.
- Der **Health Manager** überwacht den Status der Anwendungen und greift bei sich überschneidender Ressourcenverwendung ein.
- Der **Droplet Execution Agent** ermöglicht einen isolierten Betrieb der einzelnen Anwendungen, indem er deren Instanzen in sogenannten Containern verwaltet.
- Der **Blob Store** beinhaltet Anwendungscode, Buildpacks und Droplets.
- Der **Service Broker** ist für die Ressourcenversorgung und Anbindung der Dienste an die laufenden Anwendungen verantwortlich.
- Der **Message Bus** ermöglicht die interne Kommunikation zwischen den verschiedenen Elementen.
- Der **Metrics Collector** erfasst Daten der einzelnen Komponenten.
- Der **App Log Aggregator** stellt Entwicklern einen Anwendungslog zur Verfügung.

Abbildung 5 stellt den Zusammenhang der einzelnen Komponenten dar.

[63] Vgl. Cloud Foundry Foundation (2015b)
[64] Vgl. Cloud Foundry Foundation (2015a)

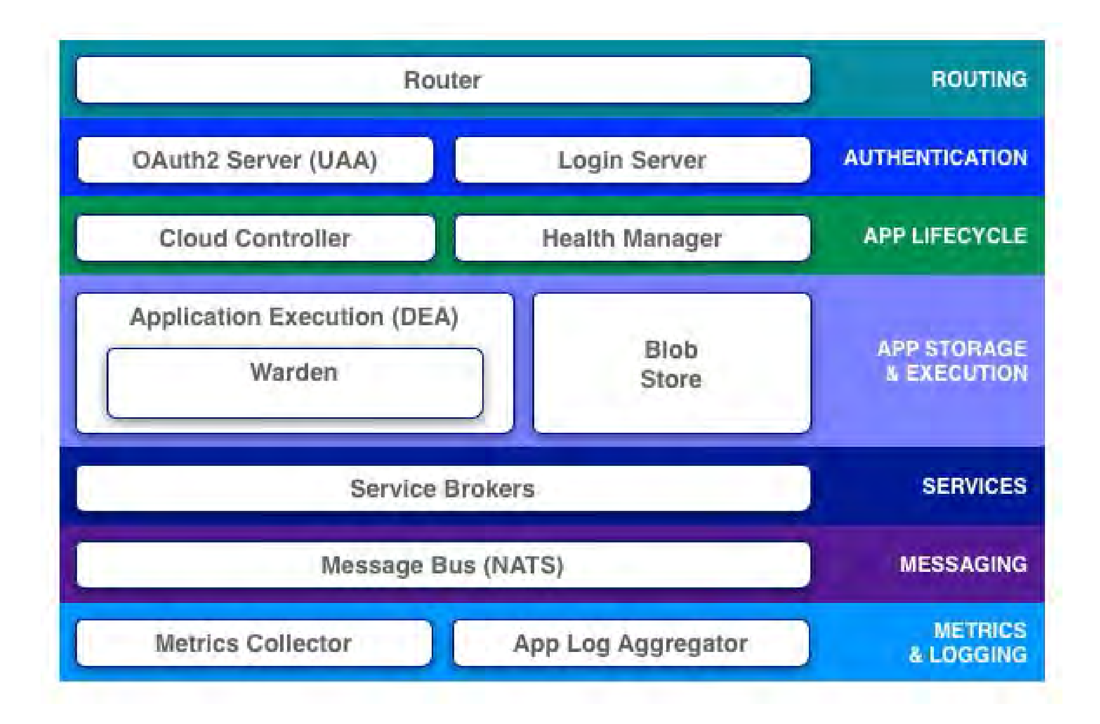


Abbildung 5: Komponenten von Cloud Foundry[65]

Für Cloud Foundry bestehen **Analyse- und Kontrollmechanismen** nur kommandozeilen-basiert. Dashboards von Drittanbietern können eingebunden werden.[66] Die Lösung ist zu dem **Betriebssystem** Linux kompatibel und kann durch die Verwendung von Maschinen-sprache auch betriebssystemunabhängig verwendet werden. Cloud Foundry empfiehlt die Nutzung der **virtuellen** Maschine vSphere von VMware, jedoch ist eine Nutzung auch ohne diese möglich.[67]

Ein **Self Service Portal** ist in der Basisversion nicht verfügbar, kann jedoch eingebunden werden. Cloud Foundry bietet eine gute **Unternehmensintegration,** da viele Programmier-sprachen für die Einbindung von Anwendungen verfügbar sind. Die Lösung ist in Bezug auf Arbeitsspeicher, Speicherplatz und VM-Instanzen **skalierbar**.[68] Hoher **Datenschutz** wird durch rollenbasierte Zugriffskontrolle gewährleistet. Die Zugangsdaten der Nutzer werden verschlüsselt übertragen.[69] Bezüglich der **Verlässlichkeit** konnten innerhalb der Recherche und der durch das Autorenteam durchgeführten Tests keine die Verlässlichkeit beeinflussen-den Auffälligkeiten festgestellt werden. Die **Benutzerfreundlichkeit** ist davon abhängig, ob ein Dashboard eines Drittanbieters eingesetzt wird. Cloud Foundry selbst bietet ausschließ-lich das kommandozeilenbasierte Management des Cloud Systems an. Eine hohe Flexibilität und große Auswahl unter verschiedenen Drittanbietern ist für die Einbindung eines Dash-

[65] Enthalten in: Cloud Foundry Foundation (2015c)
[66] Vgl. Cloud Foundry Foundation (2015d)
[67] Vgl. Cloud Foundry Foundation (2015e)
[68] Vgl. Cloud Foundry Foundation (2015f)
[69] Vgl. Cloud Foundry Foundation (2015g)

boards möglich. Der **Installationsprozess** ist äußerst komplex. Es müssen mehrere Cloud Komponenten heruntergeladen und installiert werden.[70]

Die **Innovations- und Releasegeschwindigkeit** beträgt im Durchschnitt 24 Monate, da nach der Erstveröffentlichung bisher nur im Juni 2013 eine weitere Version auf den Markt gebracht wurde.[71] Da sich in den letzten Monaten die Unterstützung durch namhafte Unternehmen deutlich erhöht hat, ist im Jahr 2015 mit einer Beschleunigung der Innovations- und Releasegeschwindigkeit sowie der Community zu rechnen.[72] Cloud Foundry stellt keinen unternehmenseigenen **Support** bereit. Die Dokumentation ist sehr ausführlich. Eine **Community** besteht in Form eines Blogs, in dem durchschnittlich einmal im Monat Neuerungen und Fehlerbehebungen veröffentlicht werden. Benutzer haben dort auch die Möglichkeit, Kommentare zu hinterlassen.[73] Fragen zum Open Source Projekt können in mehreren Google Gruppen gestellt werden. Dabei ist eine Gruppe ausschließlich für Nutzer vorgesehen.[74] In einem Wiki werden nützliche Tipps, Problemlösungen und eigenverfasste Dokumentationen von Nutzern für Nutzer veröffentlicht.[75]

6.2 CloudStack

CloudStack wurde 2010 zunächst unter der General Public License veröffentlicht. 2012 wurde CloudStack an Apache übergeben und unter die Apache Softwarelizenz gestellt. Bei CloudStack handelt es sich um eine Infrastructure as a Service Lösung. Es können sowohl Public, Private als auch Hybrid Cloud Modelle aufgebaut werden.[76] Momentan in CloudStack in der Version 4.4.2 verfügbar.[77]

Die **Kontroll- und Analysemechanismen** umfassen ein Web-Interface und eine native API, die anhand einer umfassenden Dokumentation leicht erweiterbar ist.[78] In Echtzeit können Berichte über die Ressourcennutzung erstellt werden.[79] CloudStack ist kompatibel zu verschieden **Betriebssystemen**, wie Windows, Linux und mehreren Unix-Versionen. Dabei kann vom Administrator festgelegt werden, welches Betriebssystem bei der Erstellung von neuen virtuellen Maschinen bevorzugt verwendet wird.[80] CloudStack unterstützt die **Virtuali-**

[70] Vgl. Heller, M. (2014)
[71] Vgl. Caganoff, S. (2013)
[72] Vgl. Lardinois, F. (2014)
[73] Vgl. Cloud Foundry Foundation (2015i)
[74] Vgl. Cloud Foundry Foundation (2015i)
[75] Vgl. Cloud Foundry Foundation (2015j)
[76] Vgl. Sabharwal, N. (2013)
[77] Vgl. Apache Software Foundation (2014 b)
[78] Vgl. Apache Software Foundation (2014 c)
[79] Vgl. Apache Software Foundation (2014 a)
[80] Vgl. Apache Software Foundation (2014 a)

sierungstechnologien Citrix XenServer, Citrix Xen Cloud Platform, OracleVM, KVM und VMware
vSphere.[81]

CloudStack bietet ein vollständiges **Self Service Portal** mit einem Web basierten User Interface.[82] Es existieren viele Anwendungsfälle, bei denen CloudStack in **Unternehmen integriert** wurde.[83] Das User Interface ist mit den meisten aktuellen Internet Browsern kompatibel und kann leicht in existierende Portale eingebunden werden. Außerdem kann das Aussehen der Benutzeroberfläche individuell geändert und somit an die Corporate Identity des Unternehmens angepasst werden.[84] In den Anwendungsfällen wird die gute **Skalierbarkeit** von CloudStack deutlich. TomTom, die in ihrem Unternehmen eine Private Cloud mit CloudStack aufgebaut haben, loben, dass innerhalb weniger Tage komplette virtuelle Umgebungen verfügbar gemacht werden. Das Unternehmen betreibt mehr als 150 physische Server mit CloudStack.[85] Laut Chip Childers von ASF ist die größte bekannte Anwendung von CloudStack für die Verwaltung von über 40.000 Clients zuständig.[86] CloudStack beschäftigt ein Security-Team, das sich mit den Themen **Datenschutz und Datensicherheit** auseinandersetzt.[87] Zum Schutz der Daten wird eine *secure single sign on* Nutzerverwaltung verwendet. CloudStack definiert hierfür drei Rollen:

- Der *Administrator* verwaltet die physischen und virtuellen Ressourcen.
- Der *Domain Admin* verwaltet virtuellen Ressourcen innerhalb einer Domain.
- Der *User* verwaltet seine eigenen virtuellen Ressourcen.[88]

Zum Schutz vor Angriffen von außen beinhaltet jede virtuelle Maschine eine Firewall. CloudStack sorgt für eine automatisierte Verteilung der Ressourcen über die physische Infrastruktur, wobei strenge Richtlinien für den Ausgleich von Belastungen befolgt werden. Außerdem wird der Status der virtuellen Maschinen durchgehend überwacht und eventuelle Fehler behoben. Dadurch kommt es selten zu Ausfallzeiten und das System kann als besonders **verlässlich** angesehen werden.[89] CloudStack bietet ein webbasiertes User Interface an,[90] das sehr **benutzerfreundlich** ist.[91] In Abbildung 6 ist ein Screenshot des CloudStack Dashboards abgebildet.

[81] Vgl. Lando, G. (2014)
[82] Vgl. Apache Software Foundation (2014 c)
[83] Vgl. Apache Software Foundation (2014 d)
[84] Vgl. Apache Software Foundation (2014 a)
[85] Vgl. Apachecloudstack (o. J.), S.3
[86] Vgl. Yegulalp, S. (2013)
[87] Vgl. Apache Software Foundation (2014 e)
[88] Vgl. Apache Software Foundation (2014 a)
[89] Vgl. Apache Software Foundation (2014 a)
[90] Vgl. Apache Software Foundation (2014 c)
[91] Vgl. Sourceforge (2015)

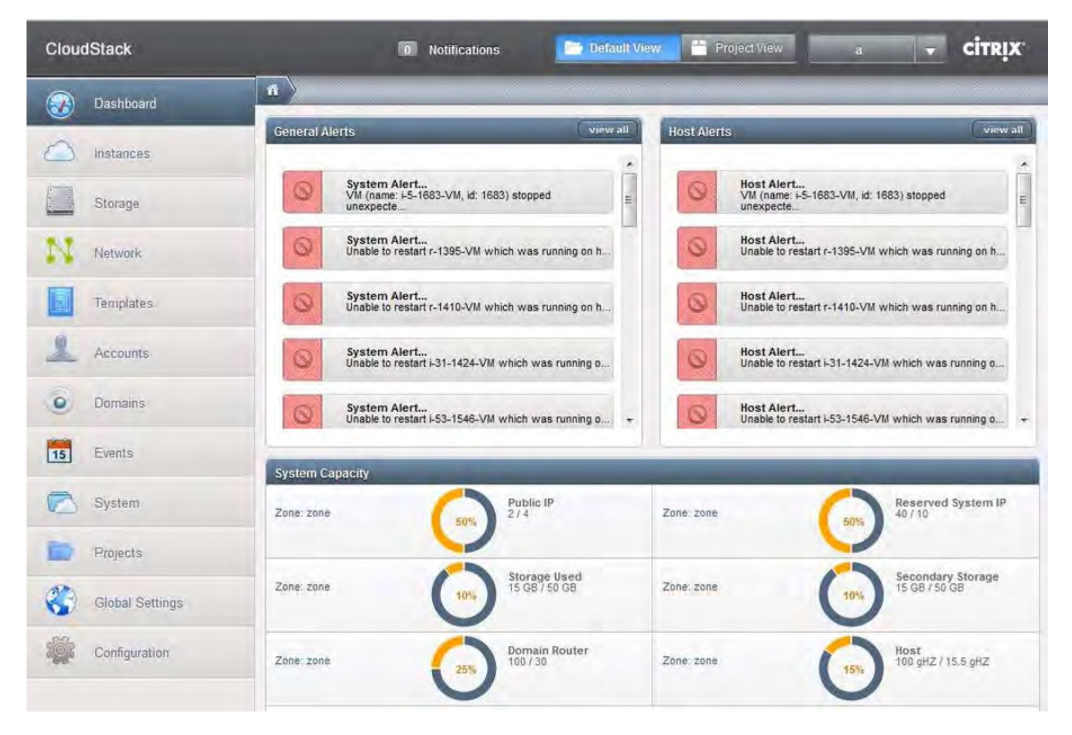


Abbildung 6: CloudStack Dashboard[92]

Der **Installationsprozess** wird von Nutzern als sehr komplex und teilweise instabil beschrieben. Für Einzelpersonen und kleinere Unternehmen ist es daher nicht zu empfehlen.[93]

CloudStack veröffentlicht durchschnittlich alle zwölf Monate ein neues **Release**.[94] CloudStack bietet keinen eigenen kommerziellen **Support**. Eine ausführliche Dokumentation ist allerdings auf der Webseite verfügbar.[95] Die CloudStack **Community** agiert auf verschiedenen Plattformen. Es gibt einen *Internet Relay Chat*, eine Mailing Liste und einige Tutorials. CloudStack hat kein eigenes Forum für Anwenderfragen.[96]

6.3 Eucalyptus

Eucalyptus ist eine Open Source Private Cloud Software, die auf unternehmensinternen Servern läuft. Der Name Eucalyptus ist ein Acronym und bedeutet *Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems.* Eucalyptus eignet sich für den Aufbau einer Infrastructure as a Service Umgebung und ist unter der General Public Licence verfügbar.[97] Die erste Version von Eucalyptus wurde 2008 im Rahmen eines Forschungspro-

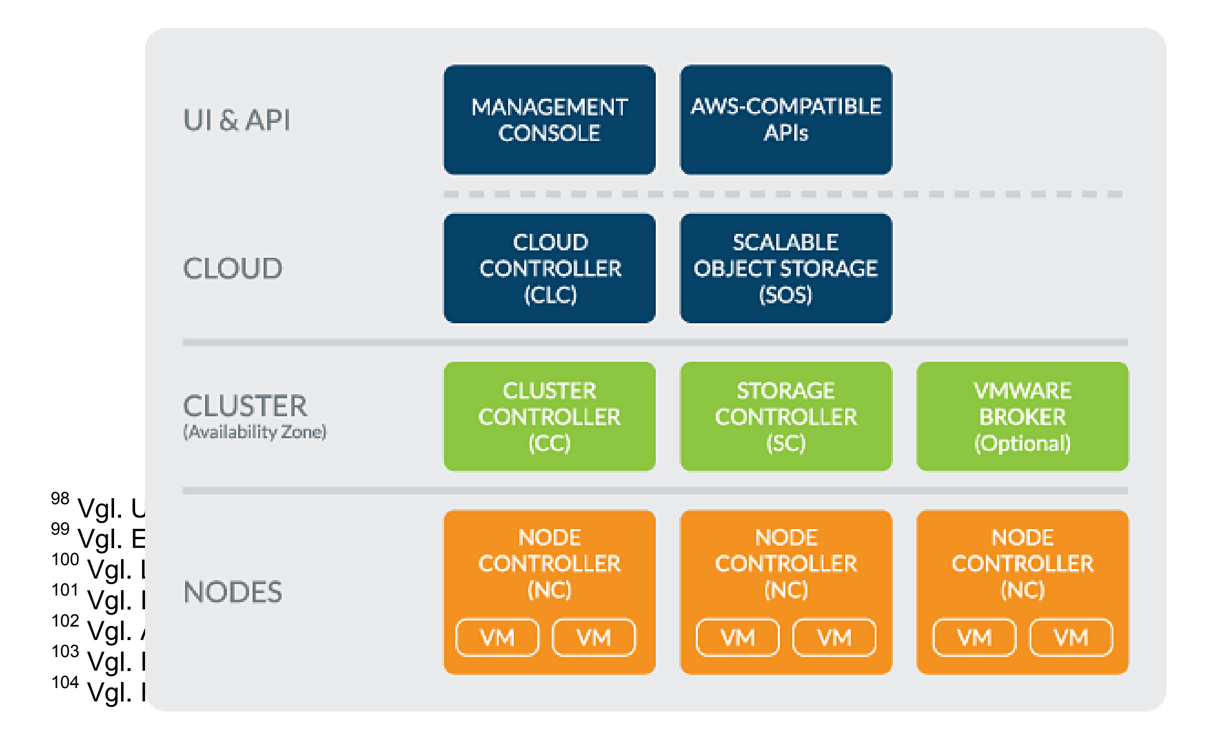
[92] Enthalten in: Socializedsoftware (2012)
[93] Vgl. Koster, M. (2013)
[94] Vgl. Apache Software Foundation (2014 f)
[95] Vgl. Apache Software Foundation (2014 c)
[96] Vgl. Huang, A. / Brockmeier, J. (2013)
[97] Vgl. Friedrich, A. (2012)

jektes der *University of California, Santa Barbara,* entwickelt und herausgegeben.[98] 2009 wurde das Unternehmen Eucalyptus Systems Inc. gegründet und im September 2014 von Hewlett-Packard gekauft. Die aktuelle Version von Eucalyptus ist 4.0.2 (Okt. 2014).[99] Eucalyptus ist kompatibel zu Amazon Web Services (AWS) und wird als Private Cloud Version dessen angesehen.[100]

Eucalyptus besteht aus fünf Basiskomponenten und einer optionalen Komponente.

- Der **Cloud Controller** dient zur Kontrolle der gesamten Cloud und ist der zentrale Zugriffspunkt für Administratoren und User.
- Der **Scalable Object Storage** ist der zu AWS Simple Storage Service (S3) kompatible Speicherdienst von Eucalyptus. [101]
- Der **Cluster Controller** ist für die lastabhängige Verteilung der virtuellen Maschinen zuständig.
- Der **Storage Controller** verwaltet die persistente Speicherung von Daten. Die Daten können unabhängig von der Aktivität der Instanz, beispielsweise einer virtuellen Maschine, dauerhaft gespeichert werden.[102]
- Der **VMware Broker** ist eine optionale Komponente und stellt eine zu AWS kompatible Schnittstelle für VMware Umgebungen zur Verfügung. Der VMware Broker wandelt Eucalyptus Machine Images zu VMware Virtual Disks um, die für VMware Umgebungen nötig sind. [103]
- Der **Node Controller** steuert die Instanzen der virtuellen Maschinen und die Endpunkte des virtuellen Netzwerks.[104]

Abbildung 7 stellt die verschiedenen Komponenten von Eucalyptus dar.



[98] Vgl. U
[99] Vgl. E
[100] Vgl. I
[101] Vgl. I
[102] Vgl. /
[103] Vgl. I
[104] Vgl. I

Abbildung 7: Komponenten von Eucalyptus[105]

Eucalyptus verfügt über umfassende Management Tools, die zur **Kontrolle und Analyse** der Cloud dienen. Die Management Plattform von Eucalyptus ermöglicht die Administration der virtuellen Maschinen, des Speichers, der CPU-Auslastung und der Nutzergruppen.[106] Die aktuelle Version von Eucalyptus kann mit verschiedenen **Betriebssystemen** von Linux und Windows verwendet werden. Eucalyptus unterstützt die **Virtualisierungstechnologien** VMware ESXi und KVM.

Über ein **Self Service Portal** können Dienstleistungen hinzugebucht werden.[107] Laut Erfahrungsberichten lässt sich Eucalyptus sehr gut in das **Unternehmen integrieren**.[108] Eucalyptus zeichnet sich durch eine hohe **Skalierbarkeit** aus, beispielsweise in Form der automatischen Skalierung von Servern.[109] Eucalyptus stellt unterschiedliche Methoden zum **Datenschutz** und zur **Datensicherung** bereit. Es können Security Groups angelegt werden und die einzelnen Komponenten haben redundante Sicherungen.[110] Die **Verlässlichkeit** von Eucalyptus wurde in verschiedenen Case Studies ebenfalls sehr positiv bewertet,[111] im Test mit der Demoversion konnten keine Ausfälle festgestellt werden. Ein intuitives Dashboard erhöht die **Benutzerfreundlichkeit** und gibt einen ausführlichen Überblick über den aktuellen Status von beispielsweise gestarteten Prozessen, Nutzern, Gruppen und verwendeten Images. Abbildung 8 zeigt einen Screenshot des Eucalyptus Dashboards der Demo Version Community Cloud.

[105] Enthalten in: Eucalyptus Systems, Inc. (2014e)
[106] Vgl. Wadia, Y. (2012), S. 32
[107] Vgl. Eucalyptus Systems, Inc. (2014f)
[108] Vgl. Eucalyptus Systems, Inc. (2014g)
[109] Vgl. Spencer, H. (2014)
[110] Vgl. Eucalyptus Systems, Inc. (2014h)
[111] Vgl. Eucalyptus Systems, Inc. (2014g)



Abbildung 8: Eucalyptus Dashboard, Demo Version Community Cloud[112]

Eucalyptus stellt eine kostenlose Demoversion, die Eucalyptus Community Cloud, zur Verfügung. Um mit dieser Software zu experimentieren, sind kein Download und keine **Installation** erforderlich – Nutzer können die Demoversion über das Webportal testen.

Eucalyptus veröffentlicht zweimal jährlich eine neue **Version**[113] und wird in drei Abonnements angeboten: Starter, Standard und Premium Pakete sind verfügbar.[114] Diese Arbeit beschränkt sich auf das von Eucalyptus angebotene Premium Paket. Das Premium Paket ist für 299$ monatlich pro Server erhältlich. Der **Support** ist rund um die Uhr, sprich 24/7, erreichbar. Die erste Reaktionszeit auf Tickets beträgt zwei Stunden. Eucalyptus hat keine **Community** in Form einer eigenen Nutzer-zu-Nutzer Plattform. Eucalyptus bietet jedoch eine Mailing Liste und ein Kontaktformular an, über welche technische Fragen innerhalb von Minuten beantwortet werden. Das Projektteam hat die Kontaktmöglichkeiten von Eucalyptus getestet und unmittelbar die gewünschten Auskünfte erhalten. In zahlreichen IT-Foren wird über Eucalyptus diskutiert und umfangreiche Trainings, Dokumentationen und Präsentationen werden von Eucalyptus bereitgestellt.

[112] Enthalten in: Eucalyptus Systems, Inc. (2014i)
[113] Drilling, T., Ostler, U. (2013), S.2
[114] Vgl. Eucalyptus Systems, Inc. (2014j)

6.4 OpenNebula

OpenNebula ist eine Infrastructure as a Service Software, mit der aus vorhandenen Rechenzentren eine Private Cloud Umgebung aufgebaut wird. OpenNebula unterstützt die Programmiersprachen Java und Ruby.[115] Es kann auch zum Aufbau von Public und Hybrid Cloud Lösungen eingesetzt werden.[116] OpenNebula wurde im März 2008 von Ignacio Llrente und Rubén Montero veröffentlicht und ist aktuell in der Version 4.10.2 unter der Apache Lizenz verfügbar.[117]

Die Architektur ist in drei verschiedene Module strukturiert, die sämtliche Komponenten enthalten: [118]

- Das erste Modul **Tools** enthält Funktionen für den Anwender und Administrator.
- Das zweite Modul **Core** enthält Anfrageverarbeitung und Kontroll-Ressourcen.
- Das dritte Modul **Drivers** enthält Treiber, Funktionen zum Datentransfer und die Steuerung der virtuellen Maschinen.

Zur **Kontrolle und Analyse** der physischen und virtuellen Maschinen kann eine Kommandozeile oder eine API verwendet werden. Ein Web Interface wird über die Plattform von Sunstone angeboten. Abhängig von der aktuellen Auslastung werden Arbeitslast und Ressourcen verwaltet und automatisch durch einen Lease Manager kontrolliert.[119] Informationen über den Status der Ressourcen, die laufenden virtuellen Maschinen und das genutzte Netzwerk lassen sich über die Kommandozeile auslesen bzw. auf dem Dashboard anzeigen.[120]

OpenNebula unterstützt das **Betriebssystem** Linux und ist in der Version Ubuntu bereits im Installationspaket enthalten.[121] Es werden Xen, KVM und VMware ESX **Virtualisierungstechnologien** unterstützt.[122]

OpenNebula bietet den Nutzern ein grafisches **Self Service Portal**, in Form eines *Marketplace*, in der die möglichen Cloud Anwendungen aufgelistet sind und vom Anwender bestellt werden. Dieses Interface wird, genauso wie die Plattform für den Administrator, von OpenNebula Sunstone bereitgestellt.[123] Architektur, Schnittstellen und Komponenten sind flexibel anpassbar und machen es so möglich OpenNebula in das **Unternehmen zu integrieren** und

[115] Vgl. Wind, S. (2011)
[116] Vgl. Buest, R. (2012)
[117] Vgl. OpenNebula (2014 a)
[118] Vgl. Wind, S. (2011)
[119] Vgl. Buest, R. (2012)
[120] Vgl. Wolters, F. u.a.. (2009)
[121] Vgl. Wolters, F. u.a.(2009)
[122] Vgl. Buest, R. (2012)
[123] Vgl. OpenNebula (2014 b)

mit anderen Systemen oder Management Tools zu verknüpfen.[124] Durch den modularen Aufbau der Systemarchitektur, lässt sich diese flexibel durch Plug-ins erweitern und an andere Produkte koppeln. OpenNebula wird für kleine und große Unternehmen empfohlen.[125] Viele Großunternehmen und öffentliche Institutionen nutzen OpenNebula bereits zur Verwaltung ihrer Cloud Systeme.[126] OpenNebula ist flexibel und bedarfsorientiert **skalierbar**.[127] Die Erstellung von neuen virtuellen Maschinen kann durch das Klonen existierender Maschinen schnell und einfach vorgenommen werden.[128] Bis zu 500 Hosts können durch automatische Ressourcenallokation genutzt werden.[129] OpenNebula beinhaltet verschiedene **Datenschutzmaßnahmen**. Nutzergruppen werden durch den Administrator verwaltet. Ein Sicherheitsmanagement-System ist für die Authentifizierung der Nutzergruppen zuständig. Verschiedene Nutzergruppen können mit individuellen Attributen erstellt werden, um verschiedene Projekte, Abteilungen oder Positionen zu repräsentieren. Die Mitglieder einer Nutzergruppe erhalten Zugang zu geteilten Ressourcen.[130] Die Authentifizierung der Nutzer erfolgt durch Passwörter, Secure Shell, RSA Schlüsselpaare und die Verwendung des Lightweight Directory Access Protocol. OpenNebula verwendet persistente Datenbanken, um seine **Verlässlichkeit** zu erhöhen. Diese Datenbanken speichern laufend Informationen über die virtuellen Maschinen, um so mögliche Fehler aufzuspüren. Im Falle einer Fehlfunktion wird das Programm neu gestartet und die laufenden VMs wiederhergestellt.[131] OpenNebula wirbt mit der intuitiven Bedienbarkeit und **Benutzerfreundlichkeit** seiner Portale.[132] Dafür verwendet es die GUI von Sunstone. Das Dashboard für OpenNebula von Sunstone ist in Abbildung 9 zu sehen.



[124] Vgl.
[125] Vgl. Wolters, F. u.a. (2009)
[126] Vgl. Wind, S. (2011)
[127] Vgl. OpenNebula (2014 b)
[128] Vgl. Buest, R. (2012)
[129] Vgl. Bud, V. (2013)
[130] Vgl. OpenNebula (2014 c)
[131] Vgl. Wind, S. (2011)
[132] Vgl. OpenNebula (2014 b)

Abbildung 9: OpenNebula Dashboard[133]

Mit der **Installation** werden alle Hauptfunktionalitäten für einen Einsatz im Unternehmen, die Speicher und die Netzwerke installiert. Die Webseite enthält eine ausführliche Installationsanleitung.[134]

Seit der ersten Veröffentlichung wurde im Durchschnitt alle 17 Monate ein neues **Release** der Software veröffentlicht.[135] OpenNebula bietet einen kostenpflichtigen **Support** an.[136] Das Premiumangebot liegt bei jährlich 900€ pro Server und umfasst einen 24/7 Support mit Antwortzeiten ab zwei Stunden.[137] Kostenlose Unterstützung erhalten die Nutzer über Dokumentationen, eine Mailing Liste und einen *Internet Relay Chat* (IRC). Die Mitglieder der **Community** sind freundlich und engagiert. Je nach Arbeitsaufwand dauert es relativ lange, bis man eine Antwort erhält.[138]

6.5 OpenQRM

OpenQRM ist ein Produkt von Qlusters, Inc. und wurde 2001 veröffentlicht.[139] Die Community Edition von openQRM steht unter der General Public License und kann in der aktuellen Version 5.2 kostenlos heruntergeladen und installiert werden.[140] OpenQRM richtet sich vor allem an Großunternehmen und Betreiber von Rechenzentren im kommerziellen Umfeld.[141]

Die Zusammensetzung der openQRM Software erfolgt nach dem Baukastenprinzip. Die Basis der Software wird durch verschiedene Plug-ins erweitert. Momentan gibt es 52 solcher Plug-ins.[142] Der Nutzer kann eigenständig über die Verwendung der Plug-ins entscheiden und dadurch die Funktionsweise der Cloud Lösung an seine Bedürfnisse anpassen.[143]

OpenQRM bietet **Kontroll- und Analysemechanismen** über drei Plug-ins an: *Nagios*, *collectd* und *i doit*. *Nagios* liefert eine automatische Überwachung, der CPU, des Speichers und des Netzwerks,[144] sowie eine Alarmierung bei Fehlern im System. *Collectd* erstellt Systemstatistiken über die Leistung und Auslastung der Cloud Lösung. *I doit* dokumentiert alle In-

[133] Enthalten in: OpenNebula (2013)
[134] Vgl. Wind, S. (2011)
[135] Vgl. OpenNebula (2014 a)
[136] Vgl. Wolters, F./ Weltermann, B./ Schnaß, P./ Heider, R. (2009)
[137] Vgl. OpenNebula (2014 d)
[138] Vgl. Bud, V. (2013)
[139] Vgl. Dhamdhere, S. (2013)
[140] Vgl. openQRM (o. J. d)
[141] Vgl. Wolters, F. u.a. (2009)
[142] Vgl. openQRM (o. J. a)
[143] Vgl. Wolters, F. u.a. (2009)
[144] Vgl. Dhamdhere, S. (2013)

formationen über die Objekte und Geschehnisse im Datencenter. OpenQRM ist kompatibel zu den **Betriebssystemen** Linux, Windows[145] und OpenSolaris.[146] Es stehen bereits vorkonfigurierte openQRM Server Pakte für Debian, Ubuntu und CentOS zur Verfügung.[147] Open QRM unterstützt sieben **Virtualisierungstechnologien**, unter anderem KVM, Citrix XenServer, VM Ware und Virtualbox.[148] So ist eine Migration von physischen zu virtuellen, genauso wie von virtuellen zu physischen und zwischen virtuellen Maschinen möglich.[149]

OpenQRM besitzt ein **Self Service Portal** mit integrierter Rechnungserstellung, in dem Nutzer nach Bedarf neue Server und Applikationen bestellen.[150] Laut der Computerwoche ist openQRM besonders gut für die **Integration ins Unternehmen** geeignet.[151] Mit den entsprechenden Plug-ins sind Schnittstellen an die Software von Drittanbieter möglich.[152] Laut dem Open Source Spezialist Plilip Dubois ist die **Skalierbarkeit** der openQRM Software sehr gut.[153] Es können problemlos komplette virtuelle Serverumgebungen bestellt werden. OpenQRM besitzt ein Plug-in, das die Nutzerrollen verwaltet und mit dem Nutzergruppen und Zugangsbestimmungen administriert werden. Durch LDAP und Active Directory werden Nutzer authentifiziert, um die **Daten** in der Cloud vor fremdem Zugriff zu **schützen**.[154] Anfallende Sicherheitsupdates sind auch in der Community Version von openQRM kostenfrei verfügbar.[155] Durch das *N-1 Failover System* kann eine größere Anzahl von physischen und virtuellen Systemen, im Falle eines Ausfalls, auf einen einzigen Standby Host umgeschaltet und gesichert werden.[156,157] Dementsprechend werden ausfallende Ressourcen umgelagert und neu gestartet, wodurch eine hohe **Verlässlichkeit** sichergestellt ist.[158] Eine solche Umlagerung kann auch zwischen virtuellen und physischen Ressourcen stattfinden.[159] Auch von Nutzern wird openQRM als stabil und verlässlich beschrieben.[160] Für die Bewertung der **Benutzerfreundlichkeit** müssen zwei Aspekte berücksichtigt werden. Aus einigen Nutzerbewertungen geht hervor, dass die Benutzerschnittstelle fehleranfällig und komplex ist. Andere

[145] Vgl. Dhamdhere, S. (2013)
[146] Vgl. openQRM (o. J. c)
[147] Vgl. Buest, R. (2012)
[148] Vgl. openQRM (o. J. c)
[149] Vgl. Wolters, F. u.a. (2009)
[150] Vgl. openQRM (o. J. a)
[151] Vgl. Radonic, A. (2011), S.3
[152] Vgl. Wolters, F. u.a. (2009)
[153] Vgl. Dubois, P. (2011)
[154] Vgl. openQRM (o. J. c)
[155] Vgl. openQRM (o. J. d)
[156] Vgl. Radonic, A. (2011), S.3
[157] Vgl. Buest, R. (2012)
[158] Vgl. Dhamdhere, S. (2013)
[159] Vgl. Radonic, A. (2011), S.3
[160] Vgl. sourceforge (2014 a)

Benutzer bewerten das Design der Schnittstelle als ansprechend und übersichtlich. Das Dashboard von openQRM ist in Abbildung 10 zu sehen.[161]

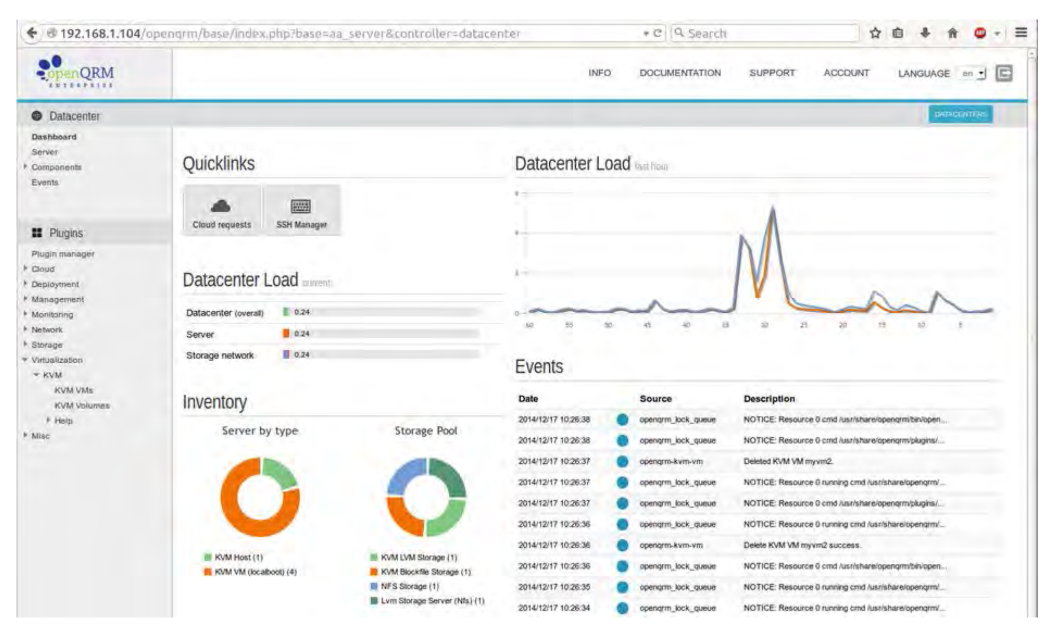


Abbildung 10: OpenQRM Dashboard[162]

Der **Installationsprozess** ist sehr umfangreich und aufwendig. Es gibt keine eindeutige Dokumentation, sodass der Nutzer die einzelnen Installationsschritte selbst recherchieren muss.[163] Für Privatnutzer und kleine Unternehmen wird der Prozess deshalb als zu aufwendig erachtet.

Open QRM hat seit der ersten Veröffentlichung im Durchschnitt alle 33 Monate ein neues **Release** veröffentlicht.[164] Die Anwender können auch ohne Support auf die Ressourcen der Anbieter, wie Dokumentation, Whitepapers und Tutorien zurückgreifen.[165] Zusätzlich bietet openQRM verschiedene kommerzielle **Supportleistungen** an, die zwischen €432 und €1920 monatlich kosten. Die Premiumvariante bietet seinen Nutzern eine 24/7 Fehlerbehebung und eine Antwortzeit von zwei Stunden.[166] Die **Community** von openQRM ist auf sourceforge.net. Die Beantwortung von Fragen in diesem Forum dauert oft bis zu einer Woche. [167] Es sind wenige Tutorials verfügbar.[168] Hilfe erhalten die Anwender über den Kontakt und den Verlauf einer Mailing Liste.[169]

[161] Vgl. Dubois, P. (2011)
[162] Enthalten in: openQRM (o. J. b)
[163] Vgl. sourceforge (2014 a)
[164] Vgl. Dubois, P. (2011)
[165] Vgl. openQRM (o.J. e)
[166] Vgl. openQRM (o. J. f)
[167] Vgl. sourceforge (2014 b)
[168] Vgl. sourceforge (2014 a)
[169] Vgl. sourceforge (2014 c)

6.6 OpenStack

OpenStack ist ein umfangreiches Software-Projekt zur Bereitstellung von Public und Private Cloud Lösungen. Hinter OpenStack steht kein Unternehmen, sondern eine Community aus mehreren Unternehmen, die das Projekt unterstützen. OpenStack wurde im Oktober 2010 in einem gemeinsamen Projekt der NASA und Rackspace initiiert. Heute gehören über 460 Unternehmen zur Community, darunter internationale Unternehmen wie Hewlett-Packard, IBM, Ubuntu, SAP und VMware.[170] OpenStack wird zum Aufbau von Infrastructure as a Service Cloud Lösungen genutzt.

Die Cloud Architektur besteht aus fünf Komponenten:

- Die erste Komponente **Compute**, auch unter dem Namen *Nova* bekannt, ist für die Verwaltung der Virtualisierung zuständig. Sie automatisiert die Abläufe der Virtual Maschine.[171]
- Die zweite Komponente **Storage** ist unterteilt in Object Storage und Block Storage:[172]
    1. OpenStack Object Storage (*Swift*) ist für die redundante Datenspeicherung verantwortlich. Diese Komponente kann unabhängig von den anderen Komponenten, zur Speicherung von Bildern oder Dokumenten genutzt werden.
    2. OpenStack Block Storage (*Cinder*) ist verantwortlich für die Ausstattung der VMs mit einem persistentem Blockspeicher. Über eine API kann der Blockspeicher mit dem Objektspeicher verbunden werden.
- Die dritte Komponente **Networking** (*Neutron*) ist für die Netzwerkverwaltung zuständig. Dies schließt die Verwaltung einzelner Clients und die Absicherung des Netzes ein.[173]
- Die vierte Komponente **Dashboard** (*Horizon*), stellt das Webinterface dar. Hier können die Nutzer über eine benutzerfreundliche GUI die verschiedenen Komponenten verwalten und haben Zugriff auf Berichte der Auslastung.[174]
- Die fünfte Komponente **Shared Services** beinhaltet fünf Dienstleistungen:[175]
    1. OpenStack Identity Service (*Keystone*) ist für die Benutzerverwaltung zuständig. Dort werden die Zugriffsmöglichkeiten der einzelnen Nutzer geregelt.
    2. OpenStack Image Service (*Glance*) stellt dem Nutzer verschiedene Images von virtuellen Maschinen zur Verfügung, welche als Vorlage zur Kompilierung neuer Instanzen verwendet werden können.

[170] Vgl. OpenStack (o.J.a)
[171] Vgl. OpenStack (o.J.b)
[172] Vgl. OpenStack (o.J.c)
[173] Vgl. OpenStack (o.J.d)
[174] Vgl. OpenStack (o.J.e)
[175] Vgl. OpenStack (o.J.f)

3. OpenStack Telemetry Service (*Ceilometer*) erfasst die Ressourcennutzung. Hiermit werden die Auslastung und die Nutzung abgefragt.
4. OpenStack Orchestration Service (*Heat*) ist zur Orchestrierung der Cloud verantwortlich. Ganze Infrastrukturen werden mithilfe von Vorlagen konfiguriert.
5. OpenStack Database Service (*Trove*) ist zur Bereitstellung einer virtuellen Datenbank in der Cloud verantwortlich.

Das Vertriebsmodell dieser Cloud Lösung ist getrieben durch verschiedene Unternehmen, die OpenStack als Basis nutzen und mittels unternehmensspezifischer Features erweitern. Dazu gehören Rackspace, Hewlett-Packard, IBM und Oracle.[176]

Abbildung 11 zeigt den Zusammenhang der fünf Hauptkomponenten von OpenStack.

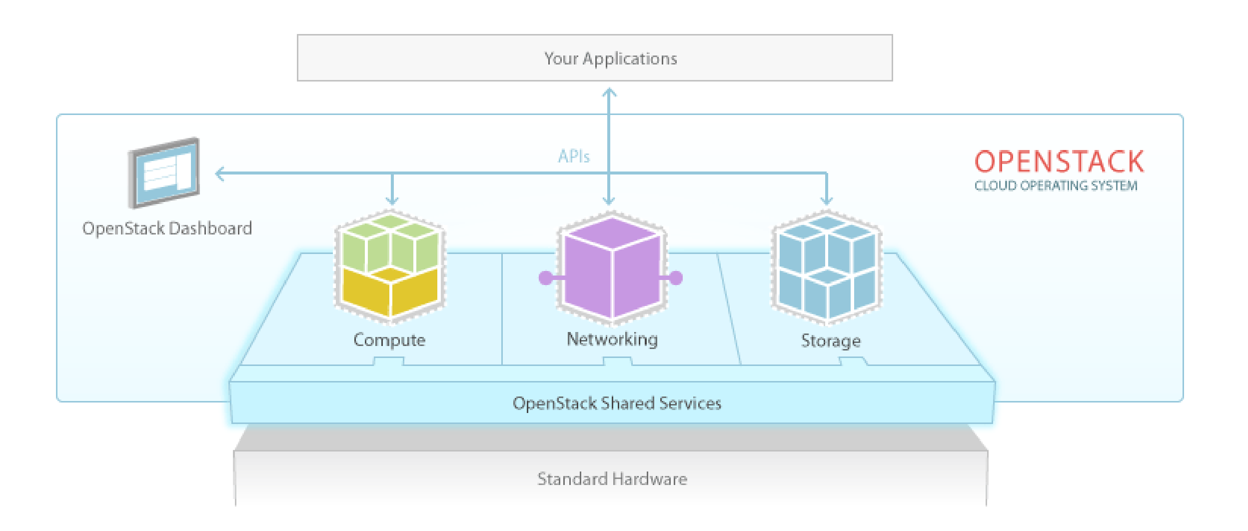


Abbildung 11: Hauptkomponenten von OpenStack[177]

Die **Kontroll- und Analysemechanismen** werden bei OpenStack durch das Dashboard bereitgestellt. Die Ressourcenauslastung und -nutzung werden genau dokumentiert und graphisch dargestellt. Über den OpenStack Identity Service wird die Nutzerverwaltung vorgenommen. Das unterstützte **Betriebssystem** ist Linux in verschiedenen Versionen, wie bspw. Ubuntu oder Debian 7.[178] Je nach Anbieter ist OpenStack auch mit weiteren Betriebssystemen kompatibel. Insgesamt besteht die **Virtualisierungstechnologie** aus elf möglichen Hypervisor, mit denen verschiedenen Features, wie bspw. der Neustart einer virtuellen Maschine, kontrolliert werden.[179]

Im Dashboard wird ein **Self Service Portal** für den Cloud Nutzer zur Verfügung gestellt, in dem alle Komponenten verwaltet werden. Für die Bewertung der **Unternehmensintegration**

[176] Vgl. OpenStack (o.J.a)
[177] Enthalten in: OpenStack (o.J.g)
[178] Vgl. OpenStack (o.J.h)
[179] Vgl. OpenStack (o.J.i)

stützt sich das Autorenteam auf den Report „Die Relevanz von OpenStack für Cloud-Sourcing“ von Büst. Laut Büst ist OpenStack einfach ins Unternehmen zu integrieren. Viele Unternehmen unterstützen die Entwicklung von Schnittstellen zu Drittsoftware und erzielen dadurch eine Standardisierung der OpenStack Software.[180] Zudem existieren über 20 Plug-ins, die zur Erweiterung der Software beitragen.[181] Die **Skalierbarkeit** von OpenStack ist einfach realisierbar: Nutzer und Speicherplatz können einfach durch die Komponenten Dashboard, Identity Service und Storage verwaltet werden. Der **Datenschutz** und die **Datensicherheit** sind unter anderem durch die Komponente Networking gegeben, die das Netzwerk durch Firewalls schützt. Zudem werden die Daten redundant gespeichert. Die **Verlässlichkeit** von OpenStack wird von Büst als sehr positiv bewertet. Dies beweist auch der große Anwenderkreis, zu dem beispielsweise PayPal, Walt Disney und Fargo zählen.[182] Die **Benutzerfreundlichkeit** wird durch ein übersichtliches Webinterface als Schnittstelle für alle Aktivitäten garantiert. Abbildung 12 zeigt das Dashboard der Demo-Version von OpenStack, auch TryStack genannt.

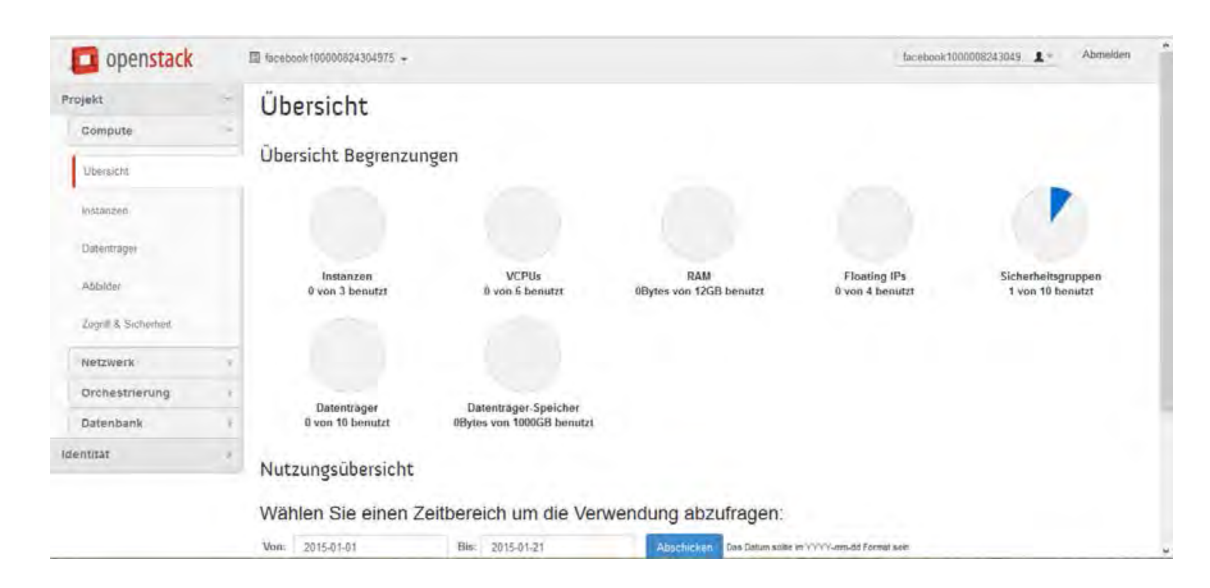


Abbildung 12: OpenStack Dashboard, Demo-Version von TryStack

Der **Installationsprozess** von OpenStack wird durch Dokumente, die auf dem Webportal zur Verfügung stehen, unterstützt. Es gibt einige Handbücher zur Installation auf verschiedenen Betriebssystemen, sowie mehrere Dokumentationen zur Anwendung der Lösung. Der Installationsprozess ist je nach Anbieter der Cloud unterschiedlich.

Die **Innovations- und Releasegeschwindigkeit** beträgt im Schnitt sechs Monate. Seit der ersten Veröffentlichung von OpenStack 2010 wurde zehn Releases veröffentlicht. Das

[180] Vgl. Büst, R (2014b)
[181] Vgl. OpenStack (o.J.j)
[182] Vgl. Büst, R (2014b)

nächste Update soll im April 2015 erfolgen.[183] Der **Support** ist vom Anbieter abhängig. OpenStack bietet User- und Entwickler-Foren an, in denen Problemstellungen und konkrete Fragen diskutiert und beantwortet werden.[184] Die OpenStack **Community** unterteilt sich in zwei Arten. Zum einen gibt es die Supporting-Community von Unternehmen die das Projekt OpenStack finanziell unterstützen und weiterentwickeln. Zum anderen gibt es die User-Community in der über 18.000 Nutzer registriert sind.[185] Auf der Community-Plattform von OpenStack gibt es die Möglichkeit konkrete Fragen zu formulieren und über Probleme zu diskutieren.

6.7 OwnCloud

Das Open Source Projekt *ownCloud* wurde vom deutschen Softwareentwickler Frank Karlitschek im Jahr 2010 initiiert.[186] Im folgenden Jahr wurde die ownCloud Inc. gegründet, die neben Open Source auch kommerzielle Softwarelösungen anbietet.[187]

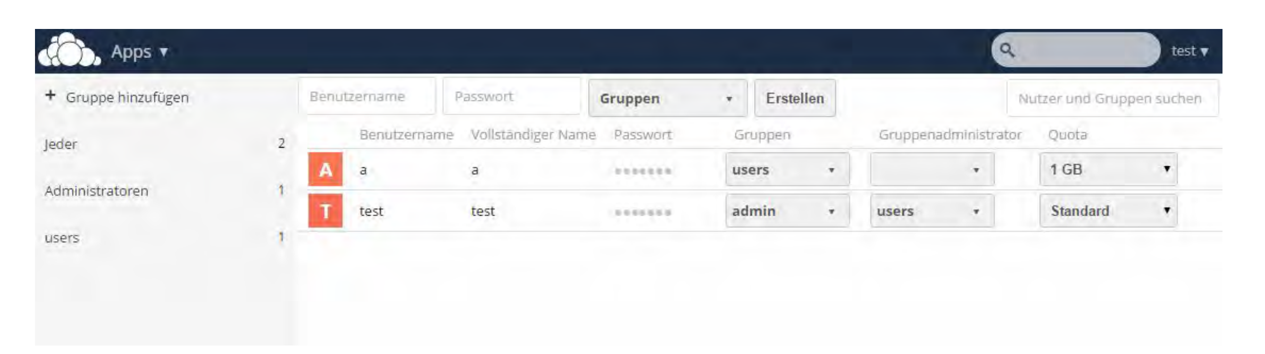
OwnCloud basiert auf dem Infrastructure as a Service Modell und wird in zwei Versionen angeboten: Einer kostenlosen Community Edition, sowie einer kostenpflichtigen Enterprise Edition.[188] Die Community Edition ist unter der Affero General Public License - einer Sonderform der General Public License - verfügbar; die Enterprise Edition unter einer kommerziellen Lizenz.[189] Diese Projektarbeit beschränkt sich auf die Community Edition von ownCloud. Die Lösung spezialisiert sich auf die Speicherung von Dokumenten in einer Private Cloud und kann automatisch mit einer lokalen Ordnerstruktur synchronisiert werden. Die aktuelle Version ownCloud 7.0.4 wurde im Dezember 2014 veröffentlicht.[190]

OwnCloud stellt eine Funktion zur **Kontrolle und Analyse** der Benutzer und des Speichers zur Verfügung. Über eine übersichtliche Weboberfläche können Benutzer hinzugefügt, sowie verschiedene Benutzergruppen angelegt werden. Eine detaillierte Überwachung des Netzwerks und der CPU-Auslastung ist nicht verfügbar. Serverseitig ist ownCloud zu den **Betriebssystemen** Windows und Linux kompatibel.[191] OwnCloud kann vom Nutzer betriebssystemunabhängig über eine Webplattform verwendet werden. Für die Betriebssysteme Linux, Windows und MacOS wird ein Desktop Client angeboten, welcher die Benutzung u.a.

[183] Vgl. OpenStack (o.J.k)
[184] Vgl. OpenStack (o.J.l)
[185] Vgl. OpenStack (o.J.m)
[186] Vgl. Karlitschek, F. (2014)
[187] Vgl. Karlitschek, F. (2014)
[188] Vgl. Vaughan-Nichols, S. J. (2013)
[189] Vgl. ownCloud Inc. (2015a)
[190] Vgl. ownCloud Inc. (2015b)
[191] Vgl. ownCloud Inc. (2015c)

durch das Wegfallen des Login-Vorgangs erleichtert.[192] Die mobilen Betriebssysteme Android und iOS werden durch eine App unterstützt.[193] Da der Nutzer mit ownCloud in der Regel keine klassische virtuelle Umgebung erschafft, werden die **Virtualisierungstechnologien** nicht bewertet.

Ein **Self Service Portal** für den Administrator ist vorhanden, welches Spracheinstellungen, das Teilen von Dateien über Servergrenzen hinweg sowie die Einbindung verschiedener externer Speicherlösungen erlaubt. Die Lösung kann über eine Schnittstelle mit Drittanbietern, wie beispielsweise Dropbox, Google Docs und Amazon S3, verbunden werden.[194] Die Lösung bietet eine gute **Unternehmensintegration**, sodass sie sich leicht mit der bestehenden Serverstruktur vereinen lässt. Nutzer und Speicherplatz sind unbegrenzt **skalierbar**, da der Anwender die Möglichkeit hat, zusätzliche Kapazitäten über Drittanbieter den.[195,196] Hoher **Datenschutz** wird serverseitig mithilfe der Encryption App – einer App zum Verschlüsseln der Daten auf dem Server – gewährleistet.[197] Die Übertragung von und zum Server ist über das Transport Layer Security Protokoll geschützt.[198] Bezüglich der **Verlässlichkeit** konnten innerhalb der Recherche und der durch das Autorenteam durchgeführten Tests keine die Verlässlichkeit beeinflussenden Auffälligkeiten festgestellt werden. Eine hohe **Benutzerfreundlichkeit** wird durch das intuitiv und einfach gestaltete Webinterface erreicht. Das Standardmenü erlaubt den Zugriff auf Dateien, Dokumente, Bilder, Kalender, Kontakte und Lesezeichen. Weitere von der Community entwickelte Apps, wie beispielsweise eine Antivirensoftware, sind in einem App-Store verfügbar und können in das benutzerspezifische Menü eingebunden werden. OwnCloud stellt zudem eine Versionierungsfunktion zur Verfügung.[199] In einer webbasierten Demoversion werden die Funktionalitäten und die Benutzeroberfläche abgebildet. Abbildung 13 zeigt einen Screenshot des ownCloud Dashboards zur Nutzerverwaltung.



[192] Vgl. ownCloud Inc. (2015c)
[193] Vgl. ownCloud Inc. (2015c)
[194] Vgl. ownCloud Inc. (2015d)
[195] Vgl. T3N-Redaktion (2012)
[196] Vgl. ownCloud Inc. (2015e)
[197] Vgl. ownCloud Inc. (2015f)
[198] Vgl. ownCloud Inc. (2015f)
[199] Vgl. ownCloud Inc. (2015d)

Abbildung 13: ownCloud Dashboard[200]

Der **Installationsprozess** gestaltet sich einfach und überschaubar. Es stehen umfangreiche Dokumentationen für User und Administratoren zur Verfügung.[201]

Die **Innovations- und Releasegeschwindigkeit** beträgt durchschnittlich acht Monate.[202] Die Community Edition bietet im Gegensatz zur Enterprise Edition, welche wahlweise einen 12/5 oder 24/7 Support anbietet, keinen **Supportservice** an. Jedoch existiert ein hochfrequentiertes **Community** Forum, in welchem Fragen innerhalb von ca. 48 Stunden beantwortet werden.[203]

6.8 Syncthing

Syncthing fokussiert sich auf die Speicherung von Dokumenten. Die Cloud Lösung, welche auf dem Infrastructure as a Service Modell basiert, wurde erstmals im Dezember 2013 vom schwedischen Softwareentwickler Jakob Borg und weiteren Mitwirkenden unter der General Public License veröffentlicht.[204] Die aktuellste Version von Januar 2015 liegt in Version 0.10.20 vor.[205]

Funktionen zur **Kontrolle und Analyse** werden über ein Webinterface bereitgestellt, auf dem Nutzer hinzugefügt und Serverinformationen, wie beispielsweise Speicher und CPU-Auslastung, verwaltet werden. Auf Serverseite ist Syncthing mit den **Betriebssystemen** Windows, MacOS, Linux und verschiedenen UNIX-basierten Betriebssystemen kompatibel.[206] Der Anwender kann sowohl über eine betriebssystemunabhängige Webplattform als auch über eine extern entwickelte App auf Syncthing zugreifen.[207] Da der Nutzer im Normalfall mit Syncthing keine klassische virtuelle Umgebung erschafft, wird die V**irtualisierungstechnologie** im Rahmen dieses Projekts nicht bewertet.

Über ein **Self Service Portal** kann der Anwender verschiedene Peer-to-Peer Verbindungen konfigurieren. Die Identifikation der Rechner erfolgt nicht über IP-Adressen, sondern über das Vergeben von IDs.[208] Diese Methode ist für den privaten Bereich durchaus geeignet, schränkt jedoch die **Integration ins Unternehmen** ein. Die Lösung ist im Hinblick auf Nut-

[200] Enthalten in: ownCloud Inc. (2015g)
[201] Vgl. ownCloud Inc. (2015h)
[202] Vgl. ownCloud Inc. (2015i)
[203] Vgl. ownCloud Inc. (2015j)
[204] Vgl. Borg, J. (2015)
[205] Vgl. Syncthing Project (2015a)
[206] Vgl, Syncthing Project (o.J. a)
[207] Vgl. Syncthing Project (o.J. a)
[208] Vgl. Syncthing Project (o.J. a)

zer- und Speicherkapazität beliebig **skalierbar**. **Datenschutz** ist durch eine verschlüsselte Dateiübertragung mithilfe des Transport Layer Security Protokolls gewährleistet.[209] Syncthing verwendet zur Übertragung der Daten ein eigenentwickeltes Peer-to-Peer Protokoll – das Block Exchange Protokoll.[210] Bezüglich der **Verlässlichkeit** konnten innerhalb der Recherche und der durch das Autorenteam durchgeführten Tests keine die Verlässlichkeit beeinflussenden Auffälligkeiten festgestellt werden. **Benutzerfreundlichkeit** wird durch das intuitiv und einfach gestaltete Webinterface erreicht. Es ist in Deutsch und in zahlreichen anderen europäischen Sprachen verfügbar. Der **Installationsprozess** ist unkompliziert. In Abbildung 14 ist das Syncthing Dashboard abgebildet.

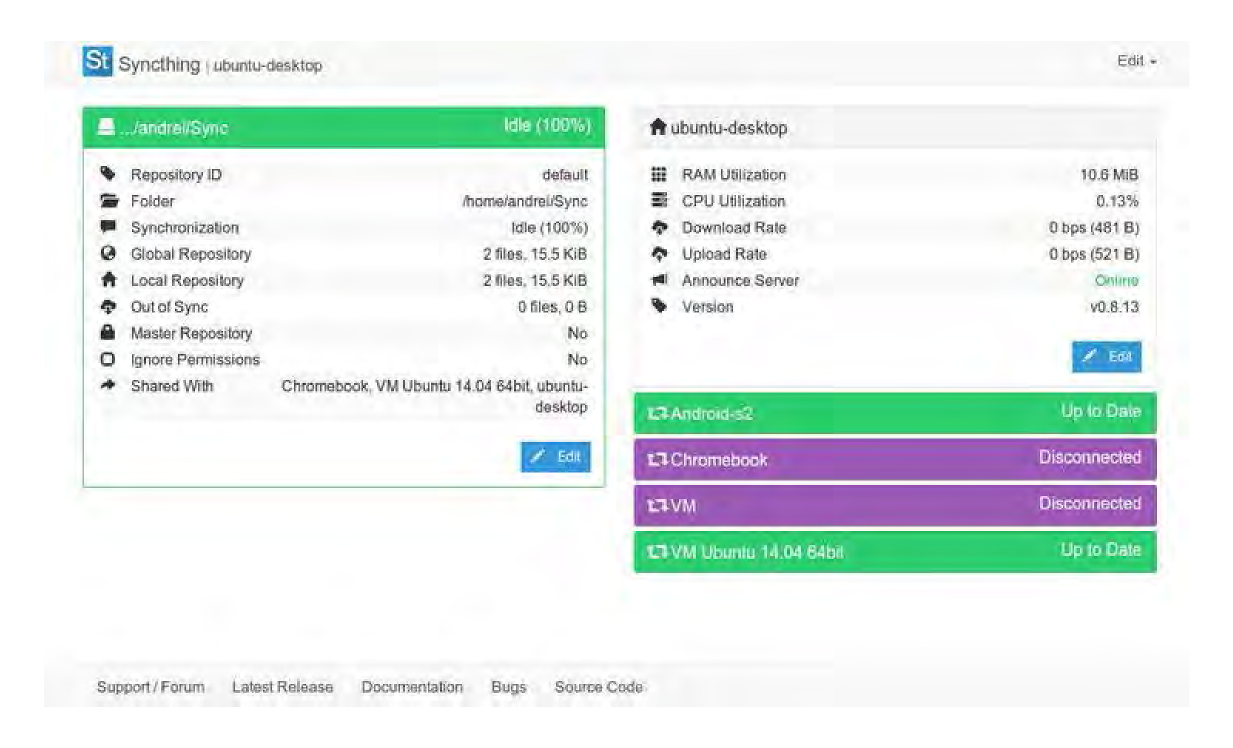


Abbildung 14: Syncthing Dashboard[211]

Seit der Erstveröffentlichung im Dezember 2013 wurde noch kein neues **Release** veröffentlicht.[212] Syncthing bietet keinen **Support**service an. Umfangreiche Dokumentationen stehen jedoch zur Verfügung.[213] Es existiert ein **Community**-Forum, welches von Nutzern und Entwicklern benutzt wird. Die Community umfasst nur wenige Mitglieder, die jedoch innerhalb von kurzer Zeit engagiert auf Fragen antworten.[214] Die Community zeigt ein leichtes Wachstum.

[209] Vgl. Syncthing Project (o. J. a)
[210] Vgl. Syncthing Project (o.J. b)
[211] Enthalten in: WebUPD (2014)
[212] Vgl. Syncthing Project (2015a)
[213] Vgl. Syncthing Project (2015b)
[214] Vgl. Syncthing Project (o.J. c)

7 Erstellung der Scorecards

In folgenden Abschnitt bewertet das Autorenteam die vorgestellten Cloud Lösungen in zwei Scorecards. Die Scorecards bilden die Grundlage für die Durchführung eines objektiven Vergleichs der verschiedenen Lösungen. In Tabelle 5 und 6 ist Scorecard I abgebildet. Tabelle 7 zeigt Scorecard II.

7.1 Scorecard I

Scorecard I	Cloud Foundry	CloudStack	Eucalyptus	Open Nebula
Funktionale Kriterien				
Kontroll- und Analysemechanismen	1	1	1	1
Nicht-funktionale Kriterien				
Management der Cloud	1	1	1	1
Skalierbarkeit	1	1	1	1
Herstellerkriterien				
Innovations- und Releasegeschwindigkeit	0	1	1	1
Support	1	1	1	1
Community	1	0	1	1
Gesamtergebnis	**5**	**5**	**6**	**6**

Tabelle 5: Scorecard I (Teil 1)

Scorecard I	OpenQRM	OpenStack	ownCloud	Syncthing
Funktionale Kriterien				
Kontroll- und Analysemechanismen	1	1	1	1
Nicht-funktionale Kriterien				
Management der Cloud	1	1	1	0
Skalierbarkeit	1	1	1	1
Herstellerkriterien				
Innovations- und Releasegeschwindigkeit	0	1	1	1
Support	1	1	1	1
Community	1	1	1	1
Gesamtergebnis	**5**	**6**	**6**	**5**

Tabelle 6: Scorecard I (Teil 2)

Das Autorenteam definiert, wie in Kapitel 5.4 „Ausschlusskriterien" beschrieben, sechs Ausschlusskriterien: Kontroll- und Analysemechanismen, Management der Cloud, Skalierbarkeit, Innovations- und Releasegeschwindigkeit, Support und Community.

Alle acht bewerteten Cloud Lösungen besitzen umfassende Funktionalitäten, um den genutzten Speicherkapazitäten, die beanspruchte CPU-Leistung und die registrierten Nutzer zu **kontrollieren** und zu **analysieren**.

Sieben der acht getesteten Cloud Lösungen stellen ein Self Service Portal bereit und lassen sich leicht in bestehende Prozesse des Unternehmens integrieren. Das **Management dieser Cloud Lösungen** kann problemlos unternehmensintern durchgeführt werden. Die Cloud Lösung *Syncthing* erhält in diesem Kriterium keine Punkte. Durch die Verwendung von Peer-to-Peer Verbindungen anstelle einer virtuellen Umgebung wird die Unternehmensintegration dieser Cloud Lösung erschwert.

Die acht analysierten Cloud Lösungen punkten alle mit ihrer flexiblen **Skalierbarkeit**. Rechenleistung, Speicherkapazitäten und Nutzergruppen können problemlos hinzu- und abgebucht werden.

Die **Innovations- und Releasegeschwindigkeit** von sechs der untersuchten Cloud Lösungen liegt bei weniger als 18 Monaten. Diesen Zeitrahmen definiert das Autorenteam als maximal zulässig, um den Anforderungen an sich ändernde Rahmenbedingungen gerecht zu werden. Die Cloud Lösungen *Cloud Foundry* und *openQRM* erfüllen dieses Ausschlusskriterium mit 24 Monaten bzw. 33 Monaten nicht.

Alle acht Cloud Lösungen bestehen den Test der **Supportleistungen**. Neben dem persönlichen Kontakt zum entwickelnden Unternehmen analysiert das Autorenteam zusätzlich die Qualität der Dokumentation und die Bereitstellung von anwendergerechten Tutorials.

Für alle acht getesteten Cloud Systeme besteht eine **Community**. Das Autorenteam bewertet in diesem Ausschlusskriterium die Aktivität der Community, die Antwortzeiten, die Qualität der Antworten und die verschiedenen Plattformen, auf denen Anwender untereinander diskutieren können. Sieben der acht Cloud Lösungen haben dabei zufriedenstellende Ergebnisse; *CloudStack* fällt im Vergleich deutlich ab und wird vom Autorenteam mit ungenügend bewertet.

Im Gesamtergebnis zeigt sich, dass vier der acht verglichenen Cloud Lösungen alle Ausschlusskriterien erfüllen. Die Cloud Systeme *Eucalyptus*, *OpenNebula*, *OpenStack* und *ownCloud* werden für einen detaillierten Vergleich auf die zweite Scorecard übertragen.

7.2 Scorecard II

Scorecard II	Gewichtung	Eucalyptus		Open Nebula		OpenStack		OwnCloud	
Funktionale Kriterien									
Kontroll- und Analysemechanismen	10,00%	5	0,5	4	0,4	5	0,5	3	0,3
Betriebssystemkompatibilität	2,00%	3	0,06	2	0,04	2	0,04	4	0,08
Virtualisierungstechnologie	2,00%	2	0,04	2	0,04	5	0,1	n/a	
Nicht-funktionale Kriterien									
Management der Cloud: Self Service Portal	5,00%	5	0,25	4	0,2	5	0,25	5	0,25
Management der Cloud: Unternehmensintegration	5,00%	4	0,2	5	0,25	5	0,25	5	0,25
Skalierbarkeit	15,00%	5	0,75	4	0,6	5	0,75	4	0,6
Benutzerfreundlichkeit	5,00%	5	0,25	5	0,25	4	0,2	5	0,25
Datenschutz & Datensicherheit	7,00%	5	0,35	5	0,35	5	0,35	5	0,35
Verlässlichkeit	7,00%	5	0,35	3	0,21	4	0,28	5	0,35
Installationsprozess	2,00%	n/a		5	0,1	3	0,06	5	0,1
Herstellerkriterien									
Innovations- und Releasegeschwindigkeit	15,00%	5	0,75	2	0,3	5	0,75	4	0,6
Support	10,00%	5	0,5	5	0,5	3	0,3	3	0,3
Community	15,00%	4	0,6	3	0,45	5	0,75	5	0,75
Gesamtergebnis	**100,00%**		**4,69**		**3,69**		**4,58**		**4,27**

Tabelle 7: Scorecard II

Nach Ausschluss von vier Cloud Lösungen durch die Scorecard I werden noch vier Cloud Lösungen durch die Scorecard II verglichen. Das sind die Private Cloud Lösungen *Eucalyptus, OpenStack, OpenNebula* und *ownCloud*. Die Auswertung des Gesamtergebnisses ergibt für *OpenNebula* 3,69 Punkte, für *ownCloud* 4,27 Punkte, für *OpenStack* 4,58 Punkte und für *Eucalyptus* die höchste Bewertung mit 4,69 Punkten.

Alle Cloud Lösungen sind sehr gut. *Eucalyptus* punktet vor allem mit nicht-funktionalen Kriterien und Herstellerkriterien. *Eucalyptus* erzielt in 9 von 13 Kategorien die Bestnote 5, und wurde nur in einer Kategorie mit 2 Punkten bewertet.

Die **Kontroll- und Analysemechanismen** von *Eucalyptus* sind überdurchschnittlich gut und werden mit 5 Punkten bewertet. Besonders positiv ist das intuitive Dashboard, über das die Ressourcen verwaltet werden, hervorzuheben. Auch *OpenStack* hat sehr gute Kontroll- und Analysemechanismen und wird mit 5 Punkten bewertet; *OpenNebula* und *ownCloud* liegen mit 4 und 3 Punkten hinten.

OwnCloud bietet als einzige der vier Lösungen eine Android und iOS-basierte Anwenderapplikation an und erhält mit 4 Punkten die beste Bewertung für die **Betriebssystemkompatibilität**. *Eucalyptus* ist mit Windows und Linux kompatibel und erhält 3 Punkte, *OpenNebula* und *OpenStack* unterstützen nur Linux und werden mit 2 Punkten bewertet.

Die meisten unterstützten **Virtualisierungstechnologien** bietet *OpenStack* mit 11 möglichen Hypervisor, wofür die Bestnote 5 vergeben wird. *Eucalyptus* und *OpenNebula* bieten jeweils nur 3 unterstütze Hypervisor an und erhalten 2 Punkte. OwnCloud funktioniert nicht mit Virtualisierungstechnologien in Form von Hypervisor und wird deshalb nicht bewertet.

Alle vier Cloud Lösungen verfügen über ein **Self Service Portal**. *Eucalyptus, OpenStack* und *ownCloud* erhalten für ihren großen Funktionsumfang die Bestnote 5; *OpenNebula* bietet ein ausgelagertes Self Service Portal der Firma Sunstone an und wird mit 4 Punkten bewertet.
OpenNebula, OpenStack und *ownCloud* bieten viele Plug-ins zur **Integration** des Systems in das **Unternehmen** an und erhalten jeweils 5 Punkte. Eucalyptus bietet eine exzellente Schnittstelle zu Amazon Web Services an, verfügt ansonsten jedoch kaum über Schnittstellen zu anderen Systemen und wird mit 4 Punkten bewertet.

Die **Skalierbarkeit** wird bei *Eucalyptus* und *OpenStack* mit 5 Punkten bewertet. *Eucalyptus* bietet die Funktion des Autoscaling an und *OpenStack* ermöglicht eine Skalierbarkeit über verschiedene Komponenten. *OpenNebula* und *ownCloud* sind ebenfalls skalierbar. Ihre Mechanismen sind im Vergleich zu *Eucalyptus* und *OpenStack* jedoch weniger umfassend.

Eucalyptus, OpenNebula und *ownCloud* erhalten für ihre **Benutzerfreundlichkeit** jeweils 5 Punkte. Im Vergleich dazu ist *OpenStack* weniger intuitiv und wird mit 4 Punkten bewertet.

Für **Datenschutz und Datensicherheit** stellen alle Lösungen umfassende Mechanismen bereit und werden mit der Höchstpunktzahl 5 bewertet.

Eucalyptus und *ownCloud* erhalten für ihre **Verlässlichkeit** jeweils 5 Punkte. *OpenStack* liegt mit 4 Punkten auf dem zweiten Rang. Das Autorenteam bewertet die Verlässlichkeit von *O-penNebula* mit 3 Punkten. Im Falle eines Fehlers werden alle Maschinen der *OpenNebula* Cloud neugestartet und es kann zu erheblichen Ausfallzeiten kommen.

Der **Installationsprozess** von *OpenNebula* und *ownCloud* ist sehr einfach und schnell und wird mit 5 Punkten bewertet. Der Installationsprozess für *OpenStack* ist kommandozeilenbasiert und komplex und erhält 3 Punkte. *Eucalyptus* stellt eine Demoversion zur Verfügung, welche über ein Webinterface getestet wird und keinen Installationsprozess erfordert. Darum wird der Installationsprozess von *Eucalyptus* nicht bewertet.

Eucalyptus entwickelt das Cloud System kontinuierlich weiter und hat einen sehr regelmäßigen **Release Zyklus** von 6 Monaten. Der Release Zyklus von *OpenStack* liegt ebenfalls bei 6 Monaten. *OpenStack* und *Eucalyptus* erhalten jeweils 5 Punkte. Der Release Zyklus beträgt bei *ownCloud* 8 Monate und wird mit 4 Punkten bewertet; *OpenNebula* veröffentlicht alle 17 Monate neue Versionen und wird mit 2 Punkten bewertet.

Beim **Support** punktet *Eucalyptus* mit einer 24/7 Verfügbarkeit und einer maximalen Reaktionszeit von zwei Stunden. Der Support wurde vom Autorenteam getestet und mit 5 Punkten bewertet. *OpenNebula* stellt, ähnlich wie *Eucalyptus*, einen umfangreichen Support zur Verfügung und erhält 5 Punkte. Der Support von *OpenStack* ist anbieterabhängig, in User- und Entwickler-Foren werden jedoch Fragen diskutiert. *OpenStack* erhält 3 Punkte in der Bewertung des Supports. Für *ownCloud* gibt es keinen Supportservice aber eine sehr umfangreiche Dokumentation. Der Support von *ownCloud* wird mit 3 Punkten bewertet.

Die **Community** ist bei *OpenStack* und *ownCloud* sehr aktiv. Beide Anbieter stellen eine Plattform für die Community zur Verfügung und werden mit 5 Punkten bewertet. *Eucalyptus* ist als Forschungsprojekt einer Universität entstanden und verfügt über eine große und aktive Entwicklergemeinschaft. Änderungsvorschläge und Verbesserungen werden von *Eucalyptus* gerne angenommen und integriert. Allerdings bietet *Eucalyptus* keine eigene Nutzerplattform an und erhält 4 Punkte für seine Community. *OpenNebula* hat ebenfalls keine eigene Nutzerplattform und zudem lange Reaktionszeiten für die Beantwortung von Anwenderfragen. Die Community von *OpenNebula* wird mit 3 Punkten bewertet.

Aus dem Vergleich mit den anderen Cloud Lösungen geht *Eucalyptus* als Testsieger hervor. *Eucalyptus* erzielte durchweg sehr gute Punktzahlen und die Tests mit der Demoversion waren erfolgreich. *Eucalyptus* ist seit sechs Jahren am Markt verfügbar und ein sehr etabliertes Produkt. Darüber hinaus ist die Community sehr aktiv. Das Projektteam empfiehlt den Einsatz von *Eucalyptus* als Open Source Private Cloud für die *.Versicherung*.

8 Fazit und Ausblick

Die unternehmensinternen IT-Abteilungen haben die Aufgabe, alle im Unternehmen ablaufenden Prozesse und anfallenden Aufgaben bestmöglich zu unterstützen. Dabei sollen sie das Unternehmen in einer sich ständig ändernden Unternehmensumwelt stärken und nach Möglichkeit einen Wettbewerbsvorteil schaffen. Um dieses Ziel zu erreichen, müssen die Mechanismen der IT-Abteilung wandelbar und schnell an die neusten Entwicklungen am Markt anpassbar sein. Denn schon Heraklit von Ephesus wusste: „Nichts ist so beständig wie der Wandel.“[215]

Der Einsatz von Cloud Lösungen im Unternehmen schafft genau dies: Cloud Lösungen bieten eine Grundlage, um abteilungsübergreifende Prozesse und von verschiedenen Organisationen geteilte Ressourcen zu managen und jedem berechtigten Nutzer Zugang zu ihnen zu

[215] Veränderungs-Consulting (2009)

verschaffen. Dabei ist die Kapazität der Ressourcen beliebig an die aktuellen Marktentwicklungen und den Unternehmensbedarf anpassbar.

Mit der Integration von Cloud Lösungen in das Unternehmen entstehen auch Risiken. Die IT-Abteilung muss vor der Einführung eines neuen Cloud Systems Antworten auf einige Fragen finden:

- Wie kann der tatsächliche Ressourcenverbrauch ermittelt werden?
- Ist die Cloud Software mit den Betriebssystemen aller Server kompatibel?
- Werden andere wichtige Systeme des Unternehmens beeinträchtigt?
- Sind die Unternehmensdaten und die Daten aller Kunden, Lieferanten und Mitarbeiter vor Angriffen von außen geschützt?
- Wird die verwendete Software regelmäßig gewartet und aktualisiert?
- Wer kann bei Fragen oder Problem mit dem Cloud System kontaktiert werden?

Das Autorenteam dieser Arbeit setzt sich intensiv mit diesen Fragen auseinander. Sie erarbeiten einen umfassenden Kriterienkatalog, der die im Unternehmen auftretenden Fragen in funktionale Kriterien, nicht-funktionale Kriterien und Herstellerkriterien unterteilt. Für jede dieser Kategorien stellen sie drei bis sieben Kriterien zusammen; von Kontroll- und Analysemechanismen über die flexible Skalierbarkeit bis hin zu einer aktiven Community und Supportleistungen.

Aus den 13 gesammelten und aufbereiteten Kriterien wählt das Autorenteam sechs Kriterien als Ausschlusskriterien aus. Das Autorenteam entwickelt in einer Gruppendiskussion eine faire Gewichtung der Kriterien zueinander, die an die unternehmensspezifische Situation des Auftraggebers *.Versicherung* angepasst ist. Eine besonders hohe Gewichtung erhalten die Kontroll- und Analysemechanismen, das Management der Cloud, die Skalierbarkeit, die Innovations- und Releasegeschwindigkeit, der Support und die Community.

Das Autorenteam untersucht den Markt für Open Source Private Cloud Lösungen und wählt acht Lösungen aus, die anhand des zuvor entwickelten Kriterienkatalogs verglichen werden. Die ausgewählten Cloud Lösungen sind: Cloud Foundry, CloudStack, Eucalyptus, OpenNebula, openQRM, OpenStack, ownCloud und Syncthing. Für die Bewertung analysiert das Autorenteam die Funktionsweisen dieser Cloud Lösungen im Detail und ermittelt Erfahrungswerte von bestehenden Anwendern. Eigene Erfahrungen sammelt das Autorenteam beim Test der Demoversionen einiger Cloud Lösungen.

Zur Bewertung und zum Vergleich der verschiedenen Cloud Lösungen erstellt das Autorenteam zwei Scorecards:

- In der ersten Scorecard werden ausschließlich die zuvor definierten sechs Ausschlusskriterien bewertet. Die Cloud Lösungen Eucalyptus, OpenNebula, OpenStack und ownCloud erfüllen alle Ausschlusskriterien und werden zur weiteren Bewertung auf die zweite Scorecard übertragen.
- In der zweiten Scorecard werden die vier übrigen Cloud Lösungen im Hinblick auf alle 13 Kriterien auf einer Skala von 1-5 bewertet. Durch Multiplikation mit der zuvor erstellten Gewichtung erhält jede der vier Lösungen einen Endpunktestand.

Auf Basis der zweiten Scorecard spricht sich das Autorenteam dafür aus, das Private Cloud System *Eucalyptus* im Unternehmen des Auftraggebers zu installieren. Eucalyptus erreicht in acht der getesteten Kriterien die Höchstpunktzahl, darunter die Kontroll- und Analysemechanismen, der Datenschutz und die Datensicherheit, die Innovations- und Releasegeschwindigkeit und die Supportleistungen. Zudem punktet es mit seiner Flexibilität: Eucalyptus ermöglicht seinen Anwendern eine bedarfsorientierte Anpassung der Speicherkapazität, der CPU-Leistung und der Nutzergruppen. Durch seine Kompatibilität zu Amazon Web Services kann Eucalyptus problemlos in eine Public Cloud umgewandelt werden oder zusammen mit einer Public Cloud zu einem Hybrid Cloud System verschmelzen. Somit erweitert Eucalyptus den Handlungsspielraum des Unternehmens hinsichtlich einer Migration zu einem anderen Cloud Einsatzmodell.

9 Literaturverzeichnis

Baun, C. u.a. (2011): Cloud Computing, Web-basierte dynamische IT-Services, 2. Aufl. Berlin/Heidelberg: Springer Verlag

Bedner, M. (2012): Cloud Computing: Technik, Sicherheit und rechtliche Gestaltung, FORUM Wirtschaftsrecht, Band 14, Kassel University

Bitcom (2010): Cloud Computing- Was Entscheider wissen müssen, Ein ganzheitlicher Blick über die Technik hinaus, Berlin: Bitcom

Dhamdhere, S. (2013): Cloud Computing and Virtualization Technologies in Libraries, IGI Global

Haun, M. (2014): Cognitive Computing, Steigerung des systematischen Intelligenzprofils, Berlin/Heidelberg: Springer Verlag

Hettl, M. (2013): Mitarbeiterführung mit dem LEAD-Navigator, Erfolgreich und wirksam führen, Wiesbaden: Springer Fachmedien

Hunziker, S. / Rihs, S. (2006): Risikoexposition bei Einsatz von Open-Source und proprietären Browsern, in: Datenschutz und Datensicherheit, S. 332-338

Hüttenegger, G. (2006): Open Source Knowledge Management, Berlin/Heidelberg: Springer Verlag

Keßler, S. (2013): Anpassung von Open-Source-Software in Anwenderunternehmen, Entwicklung und Management von Informationssystemen und intelligenter Datenauswertung, Hrsg.: Alpar, P. / Hasenkamp, U., Wiesbaden: Springer Fachmedien

Lissen, N./Brünger, C./ Damhorst, S (2014): IT-Services in der Cloud und ISAE 3402: Ein praxisorientierter Leitfaden für eine erfolgreiche Auditierung, Berlin/Heidelberg: Springer Gabler

Matros, R. (2012): Der Einfluss von Cloud Computing auf IT-Dienstleister, Eine Fallstudienbasierte Untersuchung kritischer Einflussgrößen, Wiesbaden: Gabler Verlag

Meredith, J. / Mantel, S. (2009): Project Management, A Managerial Approach, 7th Edition, o.O.: John Wiley & Sons, Inc.

Meinel, C. u.a. (2011): Virtualisierung und Cloud Computing: Konzepte, Technologiestudie, Marktübersicht, Technische Berichte des Hasso-Plattner- Instituts für Softwaresystemtechniken an der Universität Potsdam, Nr. 44, Potsdam: Universitätsverlag

Mell, P./ Grance, T. (2011): The NIST Definition of Cloud Computing, Recommendations of the National Institute of Standards and Technology Special Publication 800-145, National Institute of Standards and Technology (Hrsg.), Gaithersburg

Picot, A. / Fiedler, M. (2008): Geistiges Eigentum: Schutzrecht oder Ausbeutungstitel?, Zustand und Entwicklungen im Zeitalter von Digitalisierung und Globalisierung, Hrsg.: Depenheuer, O. / Peifer, K., Berlin/Heidelberg: Springer Verlag

Rajaraman, V. (2014): Cloud Computing, in: Resonance, o. O.: Springer India, S. 242-258

Sabharwal, N. (2013): Apache CloudStack Cloud Computing, Packt Publishing Ltd.

Sury, U. (2007): Opensource-Software und Haftung, in: Informatik-Spektrum, 30. Jg., Nr. 1, S. 45-48

Szer, B. (2013): Cloud Computing und Wissensmanagement: Bewertung von Wissensmanagementsystemen in der Cloud, Hamburg: Diplomika Verlag

Vollmer, T. (2013): Der Einstieg in die Cloud: Ein Blick auf die Technik und die Grundlagen des Cloud Computings, Hamburg: Disserta Verlag

Wind, S. (2011): Open Source Cloud Computing Management Plattformen, Einführung, Vergleich und Einsatzempfehlungen, Augsburg: Universität Augsburg, Lehrstuhl für Wirtschaftsinformatik und Systems Engineering

Winter, S. (2014): Management von Lieferanteninnovationen, Eine gestaltungsorientierte Untersuchung über das Einbringen und die Bewertung, Wiesbaden: Springer Fachmedien

Wolters, F. u.a.(2009): Cloud Computing und Open Source, Essen: Fachhochschule für Oekonomie & Management

10 Verzeichnis der Internetquellen

Amazon Web Services (o.J.): Amazon EBS – Häufig gestellte Fragen, http://aws.amazon.com/de/ebs/faqs/, Abruf: 15.01.2015

Apachecloudstack (o. J.): TomTom Speed Time-to-Market While Driving Out Costs with a Private Cloud, https://cwiki.apache.org/confluence/download/attachments/30757703/TomTom+Case+Study+Final+071613.pdf, Abruf: 20.01.2015

Apache Software Foundation (2004): Apache License, Version 2.0, http://www.apache.org/licenses/LICENSE-2.0, Abruf: 10.01.2015

Apache Software Foundation (2014 a): Apache CloudStack Features, http://cloudstack.apache.org/software/features.html, Abruf: 20.01.2015

Apache Software Foundation (2014 b): Apache CloudStack: Downloads, http://cloudstack.apache.org/downloads.html, Abruf: 20.01.2015

Apache Software Foundation (2014 c): Apache CloudStack: About, http://cloudstack.apache.org/about.html, Abruf: 20.01.2015

Apache Software Foundation (2014 d): Apache CloudStack Users, http://cloudstack.apache.org/users.html, Abruf: 20.01.2015

Apache Software Foundation (2014 e): Apache CloudStack: Security, http://cloudstack.apache.org/security.html, Abruf: 20.01.2015

Apache Software Foundation (2014 f): CloudStack's History, http://cloudstack.apache.org/history.html, Abruf: 20.01.2015

A7digital GmbH (o.J.): Die Bedeutung von Datenschutz und Datensicherheit in der Cloud, Datensicherheit, http://www.cloud-computing-deutschland.com/datenschutz.html, Abruf: 19.01.2015

Barton, T. (2012): Cloud Computing und standardisierte Schnittstellen, http://www.imittelstand.de/themen/topthema_100887.html, Abruf: 10.01.2015

Blaisdell, R. (2012): Cloud Computing ermöglicht Skalierbarkeit und Flexibilität, http://www.enterprisecioforum.com/de/blogs/rickblaisdell/cloud-computing-erm%C3%B6glicht-skalierbarkeit, Abruf: 06.12.2014

Borg, J. (2015): Nym Networks, https://nym.se/, Abruf: 20.01.2015

Brockmeier, J. / Huang, A. (2013): Community, https://cwiki.apache.org/confluence/display/CLOUDSTACK/Community, Abruf: 20.01.2015

Buest, R. (2012): Profesionelle Open-Source Lösungen für die eigene Cloud, http://clouduser.de/management/professionelle-open-source-losungen-fur-die-eigene-cloud-15397, Abruf: 20.01.2015

Bundesamt für Sicherheit in der Informationstechnik (o. J.a): Cloud Computing, https://www.bsi.bund.de/DE/Themen/CloudComputing/CloudComputing_node.html, Abruf: 10.01.2015

Bundesamt für Sicherheit in der Informationstechnik (o.J.b): Cloud Computing Grundlagen,https://www.bsi.bund.de/DE/Themen/CloudComputing/Grundlagen/Grundlagen_node.html, Abruf: 06.01.2015

Busch, D. (2012): Definition: Cloud Computing, http://blog.team-neusta.de/index.php/spektrum/definition-cloud-computing/, Abruf: 19.01.2015

Bud, V. (2013): An OpenNebula Private Cloud, http://de.slideshare.net/databuspro/an-opennebula-private-cloud-23986706?qid=45ffbcdc-8db9-4514-9d97-7f7f1cbdc69a&v=qf1&b=&from_search=3, Abruf: 20.01.2015

Büst, R. (2014a): Die Top 15 Open Source Cloud Computing Technologien 2014, http://www.crisp-research.com/die-top-15-open-source-cloud-computing-technologien-2014-2/, Abruf: 11.01.2015

Büst, R. (2014b): Crisp Research: Die Relevanz von OpenStack für das Cloud-Sourcing, http://www.crisp-research.com/die-relevanz-von-openstack-fur-das-cloud-sourcing/, Abruf: 20.01.2015

Caganoff, S. (2013): Cloud Foundry Announces New Version, Deprecates Old, http://www.infoq.com/news/2013/06/cloudfoundry_v2, Abruf: 20.01.2015

Cloud Foundry Foundation (2015a): Cloud Foundry Overview, http://docs.cloudfoundry.org/concepts/overview.html, Abruf: 20.01.2015

Cloud Foundry Foundation (2015b): Cloud Foundry Established to Advance Platform-as-a-Service, http://www.cloudfoundry.org/cloud-foundry-foundation-launch.html, Abruf: 20.01.2015

Cloud Foundry Foundation (2015c): Dashboard Single Sign-On, http://docs.cloudfoundry.org/services/dashboard-sso.html, Abruf: 20.01.2015

Cloud Foundry Foundation (2015d): Scaling Cloud Foundry, http://docs.cloudfoundry.org/concepts/high-availability.html, Abruf: 20.01.2015

Cloud Foundry Foundation (2015e): Cloud Foundry Security, http://docs.cloudfoundry.org/concepts/security.html, Abruf: 20.01.2015

Cloud Foundry Foundation (2015f): The Cloud Foundry Blog, http://blog.cloudfoundry.org/, Abruf: 20.01.2015

Cloud Foundry Foundation (2015g): Get Involved and Contribute, http://www.cloudfoundry.org/get-in/index.html, Abruf: 20.01.2015

Cloud Foundry Foundation (2015h): Cloud Foundry Wiki, https://github.com/cloudfoundry-community/cf-docs-contrib/wiki, Abruf: 20.01.2015

Cloud Foundry Foundation (2015i): Get Involved and Contribute, http://www.cloudfoundry.org/get-in/index.html, Abruf: 20.01.2015

Cloud Foundry Foundation (2015j): Cloud Foundry Wiki, https://github.com/cloudfoundry-community/cf-docs-contrib/wiki, Abruf: 20.01.2015

Drilling, T./ Ostler, U. (2013): OpenStack-Summit 2013 – oder: Was macht Red Hat in der Cloud?, http://www.datacenter-insider.de/private-cloud/hardware/articles/403392/index2.html, Abruf: 12.01.2015

Dubois, P. (2011): Infrastructure Managers, openQRM, http://de.slideshare.net/heazky/virtual-infrastructure-managers-openqrm, Abruf: 20.01.2015

Erhardt S. (o.J.): Rechtssicherheit in der Cloud: Ihre Daten bei Dropbox, iCloud, Google Drive & Co, Cloud-Anbieter und der Datenschutz: Wie sicher sind Ihre Daten?, http://www.e-recht24.de/artikel/blog-foren-web20/7115-rechtssicher-in-der-cloud-ihre-daten-bei-dropbox-icloud-google-drivea-co.html, Abruf: 19.01.2015

Essoh D./ Doubrava, Dr.C./ Münch, I. (2012): Sicherheitsempfehlungen für Cloud Computing Anbieter, http://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Mindestanforderungen/Eckpunktepapier-Sicherheitsempfehlungen-CloudComputing-Anbieter.pdf?__blob=publicationFile, Abruf: 03.01.2015

Eucalyptus Systems (2014a): Eucalyptus documentation, https://www.eucalyptus.com/docs/eucalyptus/4.0.2/index.html, Abruf: 12.01.2015

Eucalyptus Systems (2014b): Cloud Level, https://www.eucalyptus.com/eucalyptus-cloud/iaas/architecture/cloud-level, Abruf: 12.01.2015

Eucalyptus Systems (2014c): Cluster Level, https://www.eucalyptus.com/eucalyptus-cloud/iaas/architecture/cluster-level, Abruf: 12.01.2015

Eucalyptus Systems (2014d): Eucalyptus Architecture: Node Level, https://www.eucalyptus.com/eucalyptus-cloud/iaas/architecture/node-level, Abruf: 12.01.2015

Eucalyptus Systems (2014e): Eucalyptus Cloud Computing Architecture, https://www.eucalyptus.com/eucalyptus-cloud/iaas/architecture, Abruf: 12.01.2015

Eucalyptus Systems (2014f): Eucalyptus Cloud Compatibility Matrix 4.0, https://www.eucalyptus.com/eucalyptus-cloud/iaas/compatibility/4.0, Abruf: 13.01.2015

Eucalyptus Systems (2014g): Cloud Computing Case Studies, https://www.eucalyptus.com/customers/case-studies, Abruf: 19.01.2015

Eucalyptus Systems (2014h): Eucalyptus security, https://www.eucalyptus.com/resources/security, Abruf: 19.01.2015

Eucalyptus Systems (2014i): Eucalyptus community cloud, https://communitycloud.eucalyptus.com/, Abruf: 19.01.2015

Eucalyptus Systems (2014j): Buy Eucalyptus Private Cloud Subscriptions, https://www.eucalyptus.com/eucalyptus-cloud/subscriptions, Abruf: 12.01.2015

Fallenbeck, N./ Windorst, I. (2014): Sicheres Cloud Computing: Der rechtliche Rahmen, Computerwoche, http://www.computerwoche.de/a/sicheres-cloud-computing,2527898,3, Abruf: 19.01.2015

Free Software Foundation (2014): Categories of free and nonfree Software, http://www.gnu.org/philosophy/categories.en.html, Abruf: 10.01.2015

Free Software Foundation (2007): GNU General Public License, http://www.gnu.org/licenses/gpl-3.0.html, Abruf: 10.01.2015

Frey, S. u. A. (2014): SLA-Richtliniendokument für Cloud Dienstleistungen, http://www.wolke.hs-furtwangen.de/assets/files/ASLAMaaS-SLA-Richtliniendokument.pdf, Abruf: 03.01.2015

Friedrich, A. (2012): Eucalyptus Expandiert nach Deutschland, um die Akzeptanz und den Einsatz von Cloud-Services zu beschleunigen,

https://www.eucalyptus.com/news/eucalyptus-expandiert-nach-deutschland-um-die-akzeptanz-und-den-einsatz-von-cloud-services-zu-b, Abruf: 12.01.2015

Hackmann, J. (2014): Zehn Private-Cloud-Lösungen im Vergleich, http://www.computerwoche.de/a/zehn-private-cloud-loesungen-im-vergleich,2553762,2, Abruf: 03.01.2015

Heller, M. (2014): Review: Cloud Foundry brings power and polish to PaaS, http://www.infoworld.com/article/2608299/cloud-computing/review--cloud-foundry-brings-power-and-polish-to-paas.html, Abruf: 20.01.2015

Institut für Rechtsfragen der Freien und Open Source Software (o. J.): Wie unterscheidet sich „Open Source Software" von „Freeware", „Public Domain" und „Shared Source"?, http://www.ifross.org/unterscheidet-sich-open-source-software-freeware-public-domain-und-shared-source, Abruf: 10.01.2015

IT Wissen (2015): Das große Online-Lexikon für Informationstechnologie, Stichwort: Community Cloud, http://www.itwissen.info/definition/lexikon/Community-Cloud-community-cloud.html, Abruf: 19.01.2015

Karlitschek, F. (2014): Frank Karlitschek, http://karlitschek.de/about/, Abruf: 20.01.2015

Koster, M. (2013): CloudStack Single Server on Ubuntu with KVM, http://www.greenhills.co.uk/2013/08/30/cloudstack-single-server-on-ubuntu-with-kvm.html, Abruf: 21.01.2015

Lando, G. (2014): A Game of Stacks, OpenStack vs. CloudStack, http://www.getfilecloud.com/blog/2014/02/a-game-of-stacks-openstack-vs-cloudstack/#.VL7UbEeG8qy, Abruf: 20.01.2015

Lardinois, F. (2014): Cloud Foundry Foundation Launches With Support From Over 40 Companies,http://techcrunch.com/2014/12/08/cloud-foundry-foundation-launches-with-support-from-over-40-companies/, Abruf: 20.01.2015

Linthicum, D. (2014): What HP's buyout of Eucalyptus really means, http://www.infoworld.com/article/2683073/cloud-computing/what-hps-buyout-of-eucalyptus-really-means.html, Abruf: 12.01.2015

Md.Mahbub-E-Noor (2014): Virtualization and Cloud Computing, http://www.academia.edu/7121640/Cloud_Computing_and_Virtualization, Abruf: 09.01.2015

Microsoft (2015): Cloud Services Glossar: Infrastructure as a Service (IaaS), http://www.microsoft.com/de-de/cloud/glossar/infrastructure_as_a_service.aspx, Abruf: 19.01.2015

Open Source Initiative (o. J.): Welcome to The Open Source Initiative, http://opensource.org/, Abruf: 10.01.2015

Openstack (o.J.a): OpenStack Marketplace, http://www.openstack.org/marketplace/distros/, Abruf: 20.01.2015

OpenStack (o.J.b): OpenStack Compute, http://www.openstack.org/software/openstack-compute/, Abruf: 20.01.2015

OpenStack (o.J.c): OpenStack Storage, http://www.openstack.org/software/openstack-storage/, Abruf: 20.01.2015

OpenStack (o.J.d): OpenStack Networking, http://www.openstack.org/software/openstack-networking/, Abruf: 20.01.2015

OpenStack (o.J.e): OpenStack Dashboard, http://www.openstack.org/software/openstack-dashboard/, Abruf: 20.01.2015

OpenStack (o.J.f): OpenStack Shared Services, http://www.openstack.org/software/openstack-shared-services/, Abruf: 20.01.2015

OpenStack (o.J.g): OpenStack Software: The Open Source Cloud Operating System, Entnommen aus: http://www.openstack.org/software/, Abruf: 20.01.2015

OpenStack (o.J.h): Openstack Documentation: Install OpenStack, http://docs.openstack.org/, Abruf: 20.01.2015

OpenStack (o.J.i): OpenStack Wiki, Hypervisor Support Matrix, Features, http://docs.openstack.org/, Abruf: 20.01.2015

OpenStack (o.J.j): OpenStack Wiki, Neutron, Plug-Ins, https://wiki.openstack.org/wiki/Neutron, Abruf: 20.01.2015

OpenStack (o.J.k): OpenStack Wiki, Releases, https://wiki.openstack.org/wiki/Releases, Abruf: 20.01.2015

OpenStack (o.J.l): OpenStack Ask Questions, https://ask.openstack.org/en/questions/, Abruf: 20.01.2015

OpenStack (o.J.m): OpenStack, One of the fastest growing open source communities in the world, http://www.openstack.org/, Abruf: 20.01.2015

OpenNebula (2013): OpenNebula 4.2 OneFlow Service Deployment, https://www.youtube.com/watch?v=oeeKQeI6GKY, Abruf: 20.01.2015

OpenNebula (2014 a): Release Cycle, Release Schedule, http://opennebula.org/software/release/, Abruf: 20.01.2015

OpenNebula (2014 b): OpenNebula Key Features, http://opennebula.org/about/key-features/, Abruf: 20.01.2015

OpenNebula (2014 c): Release Notes, Features, http://docs.opennebula.org/4.10/release_notes/release_notes/features.html, Abruf: 20.01.2015

OpenNebula (2014 d): OpenNebula Support Subscriptions, http://opennebula.systems/support/, Abruf: 20.01.2015

openQRM (o.J. a): Enterprise-ready self-service portal. Sophisticated hybrid-cloud features. , http://www.openqrm-enterprise.com/, Abruf: 20.01.2015

openQRM (o.J. b): openQRM Community Edition, http://www.openqrm-enterprise.com/products/community-edition.html, Abruf: 20.01.2015

openQRM (o.J. c): openQRM Tour, http://www.openqrm-enterprise.com/tour.html, Abruf: 20.01.2015

openQRM (o.J. d): Edition Comparison, http://www.openqrm-enterprise.com/products/edition-comparison.html, Abruf: 20.01.2015

openQRM (o.J. e): Resources, http://www.openqrm-enterprise.com/resources.html, Abruf: 20.01.2015

openQRM (o.J. f): Support and SLAs, http://www.openqrm-enterprise.com/services/support-and-slas.html, Abruf: 20.01.2015

ownCloud Inc. (2015a): Community or Enterprise, https://owncloud.com/community-enterprise/, Abruf: 20.01.2015

ownCloud Inc. (2015b): Releases, https://github.com/owncloud/core/releases, Abruf: 20.01.2015

ownCloud Inc. (2015c): Download, http://owncloud.org/install/, Abruf: 20.01.2015

ownCloud Inc. (2015d): Features, http://owncloud.org/features/, Abruf: 20.01.2015

ownCloud Inc. (2015e): Providers, http://owncloud.org/providers/, Abruf: 20.01.2015

ownCloud Inc. (2015f): How ownCloud uses encryption to protect your data, https://owncloud.org/blog/how-owncloud-uses-encryption-to-protect-your-data/, Abruf: 20.01.2015

ownCloud Inc. (2015g): ownCloud Demo, http://demo.owncloud.org/index.php/settings/users, Abruf: 20.01.2015

ownCloud Inc. (2015h): Documentation Overview, http://doc.owncloud.org/, Abruf: 20.01.2015

ownCloud Inc. (2015i): ownCloud Server Changelog, https://owncloud.org/changelog/, Abruf: 20.01.2015

ownCloud Inc. (2015j): ownCloud Forums, https://forum.owncloud.org/, Abruf: 20.01.2015

Radonic, A. (2011): OpenQRM steuert die Private Cloud, http://www.computerwoche.de/a/openqrm-steuert-die-private-cloud,2367309,3, Abruf: 20.01.2015

Roberts, D. (2013): What is a Private Cloud? – A Tutorial, http://leverhawk.com/what-is-a-private-cloud-tutorial-20130423252, Abruf: 06.01.2015

Rouse, M. (2013): Hypervisor – Virtual Machine Monitor (VMM), http://www.searchdatacenter.de/definition/Hypervisor-Virtual-Machine-Monitor-VMM, Abruf: 09.01.2015

Ruckdeschel, R. (o. J.): Begriffsdefinition bei frei verfügbarer Software, http://www.roru.de/software/publicdomain.htm, Abruf: 10.01.2015

Spencer, H. (2014): Adding Eucalyptus Load Balancer Access Logging For Eucalyptus Cloud Users, https://www.eucalyptus.com/blog/2014/12/11/adding-eucalyptus-load-balancer-access-logging-eucalyptus-cloud-users, Abruf: 19.01.2015

Steven, J. (2014): Hypervisors: The cloud's potential security Achilles heel, http://www.zdnet.com/article/hypervisors-the-clouds-potential-security-achilles-heel/, Abruf: 09.01.2015

Socializedsoftware (2012): dashboard.150x150.jpg, http://socializedsoftware.com/wp-content/uploads/2012/02/dashboard.jpg, Abruf: 20.01.2015

Sourceforge (2014 a): openQRM - Cloud Computing Platform, Reviews, http://sourceforge.net/projects/openqrm/reviews/, Abruf: 20.01.2015

Sourceforge (2014 b): openQRM - Cloud Computing Platform, Discussion, http://sourceforge.net/p/openqrm/discussion/513343/, Abruf: 20.01.2015

Sourceforge (2014 c): openQRM - Cloud Computing Platform, Mailing List, http://sourceforge.net/p/openqrm/mailman/, Abruf: 20.01.2015

Sourceforge (2015): CloudStack - Cloud Computing Management, http://sourceforge.net/projects/cloudstack/reviews?source=navbar, Abruf: 21.01.2015

Syncthing Project (2015a): Releases, https://github.com/syncthing/syncthing/releases, Abruf: 20.01.2015

Syncthing Project (2015b): Getting started, https://github.com/syncthing/syncthing/wiki/Getting-Started, Abruf: 20.01.2015

Syncthing Project (o.J.a): Syncthing, http://syncthing.net/, Abruf: 20.01.2015

Syncthing Project (o.J.b): Block Exchange Protocol v1, https://github.com/syncthing/specs/blob/master/BEPv1.md, Abruf: 20.01.2015

Syncthing Project (o.J.c): Syncthing discussion forum, https://discourse.syncthing.net/, Abruf: 20.01.2015

T3N-Redaktion (2012): OwnCloud: Open Source Dropbox Alternativ auf eigenem Server, http://t3n.de/news/owncloud-dropbox-alternative-402069/, Abruf: 20.01.2015

UC Santa Barbara Office of Technology & Industry Alliances (2014): Eucalyptus Systems, http://tia.ucsb.edu/about-tia/success-stories/eucalyptus-systems/, Abruf: 12.01.2015

Vaughan-Nichols, S.J. (2013): Want a call where you call the shots? Consider ownCloud, http://www.zdnet.com/article/want-a-cloud-where-you-call-the-shots-consider-owncloud/, Abruf: 20.01.2015

Veränderungs-Consulting (2009): Zitate, Sprüche und Weisheiten zu Veränderung und Wandel, http://www.veraenderungs-consulting.de/Zitate/zitate.html, Abruf: 21.01.2015

VMware (2011): VMware Delivers Cloud Foundry, The Industry's First Open PaaS - See more at: http://www.vmware.com/company/news/releases/cloud-foundry-apr2011, Abruf: 20.01.2015

Wadia, Y. (2012): The Eucalyptus Open-Source Private Cloud, http://media.cloudbook.net/pdf/the-eucalyptus-open-source-private-cloud.pdf, Abruf: 12.01.2015

WebUPD (2014): SYNCTHING: OPEN SOURCE BITTORRENT SYNC ALTERNATIVE, http://www.webupd8.org/2014/06/syncthing-open-source-bittorrent-sync.html, Abruf: 20.01.2015

Yegulalp, S. (2013): Apache CloudStack 4.2, Easier upgrades, faster installation, better scaling, http://www.infoworld.com/article/2612327/iaas/apache-cloudstack-4-2--easier-upgrades--faster-installation--better-scaling.html, Abruf: 20.01.2015



Open Source Lösungen zur Spracherkennung

Schriftliche Ausarbeitung
im Rahmen der Lehrveranstaltung „Integrationsseminar“
für das Kompetenzzentrum Open Source (KOS)

Vorgelegt von

Dominic Lesinski, Jana Katherina Piller,
Sabine Schmähl, Juliane Steinacker, Marina Zistler

am 26.01.2015

Fakultät Wirtschaft
Studiengang Wirtschaftsinformatik
WI2012I

Inhaltsverzeichnis

Abkürzungsverzeichnis

API	=	Application Programming Interface
CMU	=	Carnegie Mellon University
DAPRA	=	Defense Advanced Research Projects Agency
GALE	=	Global Autonomous Language Exploitation
HTTP	=	Hypertext Transfer Protocol
URL	=	Uniform Resource Locator
USB	=	Universal Serial Bus
WER	=	Word-Error-Rate
XML	=	Extensible Markup Language

Abbildungsverzeichnis

1 Einleitung

1.1. Problemstellung

Gerätesteuerung durch menschliche Sprache, Verschriftlichung von Berichten ohne manuelle Aufschriebe oder lästiges Tippen – was vor Jahren noch kaum realisierbar schien, findet nun verstärkt Eingang in den Alltag der Menschen und unterstützt vor allem Unternehmen dabei, Abläufe im Tagesgeschäft zu beschleunigen und zu verbessern.

Das Forschungsgebiet der Spracherkennung, welche eine einfache, schnelle und bequeme Erfassung von Informationen ermöglicht, konnte in den vergangenen Jahren erhebliche Fortschritte verzeichnen und gewinnt durch die kontinuierliche Weiterentwicklung bestehender Software nun immer mehr an Bedeutung. Auch die .Versicherung als Kommunal- und Spezialversicherer plant, sich die Spracherkennung für eine Applikation speziell für Tablets zu Nutzen zu machen. Jene soll im Außendienst Verwendung finden und dazu dienen, Schadensberichte durch Diktierung zügig vor Ort erfassen zu können. Ein Schadensbericht umfasst typischerweise mehrere Zeilen Text; werden diese über ein Tablet eingegeben, führt dies zu beträchtlichen Zeitverzögerungen. Um diesen Vorgang zu beschleunigen, soll bei der .Versicherung zukünftig eine Spracherkennungssoftware, die Sprache automatisch in Text umwandelt, eingesetzt werden.

Anbieter wie Apple und Google stellen eine derartige Software kostenfrei zur Verfügung. Die Problematik besteht jedoch darin, dass jegliche Aufnahmen zur Umwandlung an den Server des Anbieters geschickt werden. Dadurch können der Datenschutz und die Privatsphäre des Kunden nicht mehr gewahrt werden. Aufgrund dessen soll nach einer alternativen Software gesucht werden, bei der kein externer Anbieter Zugang zu diesen sensiblen Daten erhält.

1.2. Zielsetzung

Zunächst soll eine Übersicht über auf dem Markt verfügbare Open Source - Spracherkennungssoftware erstellt werden. Jene Open Source - Lösungen und eine kommerzielle Lösung werden anschließend anhand eines erstellten Kriterienkataloges analysiert und die beiden am besten geeigneten Open Source - Lösungen ermittelt.

Im Anschluss daran gilt es, diese in einem eigens erstellten Prozess zu testen. Idealerweise erweist sich dabei eine der Lösungen als passend für die Belange der .Versicherung.

1.3 Vorgehensweise

Zu Beginn werden verschiedene Open Source - Lösungen in einer Liste zusammengestellt. Anhand eines Anforderungsprofils wird eine erste Vorauswahl getroffen. Im nächsten Schritt erfolgt die Erarbeitung eines detaillierten Kriterienkataloges, der verschiedene Kategorien wie Genauigkeit, Leistungsanforderungen, zusätzliche Funktionen, sowie Support umfasst. Eine entsprechende Gewichtung der Kriterien wird in Abstimmung mit der .Versicherung vorgenommen.

Anschließend werden alle Software-Lösungen einschließlich einer dem Vergleichszweck dienenden kommerziellen Lösung anhand der gesammelten Kriterien analysiert und bewertet. Nachdem zwei als geeignet befundene Open Source - Lösungen herausgefiltert wurden, wird zunächst ein Testprozess für den praktischen Einsatz der Software definiert. Dieser wird sowohl auf die gefiltere Open Source - Software als auch auf die kommerzielle Lösung angewandt, um einen besseren Vergleich zu käuflich erwerblichen Produkten herzustellen. Schließlich werden die Ergebnisse präsentiert und eine Empfehlung hinsichtlich einer den Anforderungen der .Versicherung entsprechenden Spracherkennungssoftware ausgesprochen.

1.4. Eigene Beiträge

Der Eigenteil der Projektgruppe an dieser wissenschaftlichen Arbeit umfasst insbesondere die Recherche über aktuell auf dem Markt verfügbare Spracherkennungsprogramme, die Erstellung des Anforderungsprofils und des Kriterienkataloges, sowie die Gewichtung der einzelnen Kriterien. Des Weiteren führt die Projektgruppe die Filterung der verschiedenen Programme anhand der ausgearbeiteten Kriterien und deren Bepunktung durch. Die beiden Spracherkennungsprogramme, die bei der Analyse die beste Bewertung erhalten, werden von der Projektgruppe installiert und eigenhändig getestet. Der hierzu erforderliche Testprozess wird zuvor von der Projektgruppe definiert und im Anschluss durchgeführt. Schlussendlich umfassen die eigenen Beiträge die Präsentation der Ergebnisse und das Aussprechen einer Empfehlung für die .Versicherung.

1.5. Vorbemerkungen

Aufgrund der hohen Aktualität des Themas wird nicht nur auf wissenschaftliche Quellen, sondern auch auf gesammeltes Wissen von Nutzern der verschiedenen Spracherkennungsprogramme, beispielsweise in Foren, Bezug genommen.

2 Theoretische Grundlagen

Im Folgenden werden die theoretischen Grundlagen eines Spracherkennungssystems erläutert. Dafür werden wichtige Begriffe definiert. Nach einer Ausführung über die historische Entwicklung der Spracherkennung wird auf die technischen Grundlagen eingegangen. Dieser Abschnitt soll die grundsätzlichen Vorgänge verdeutlichen, um ein Grundverständnis zu schaffen. Da der Fokus auf der Anwendung und nicht der Programmierung von Spracherkennungssoftware liegt, wird auf die technischen Grundlagen soweit eingegangen, dass ein Grundverständnis vermittelt wird.

Im Folgenden werden dann die Einsatzbereiche und Kategorien von Spracherkennungssoftware aufgezeigt. Ein weiteres Kapitel widmet sich den Schwächen automatischer Spracherkennung. Das letzte Kapitel befasst sich schließlich mit dem Kriterienkatalog, anhand dessen die verschiedenen Software-Systeme verglichen werden.

2.1. Begriffsdefinitionen

Sprache wird vom Duden als "Fähigkeit des Menschen zu sprechen“[1] bezeichnet. Diese Fähigkeit ist die grundlegendste und wichtigste für die Kommunikation zwischen Menschen und den Aufbau zwischenmenschlicher Beziehungen[2].

Auch für den wissenschaftlichen Fortschritt und kulturelle Interaktionen ist Sprache wichtig – mit ihr werden Gedanken und Meinungen ausgetauscht, Wünsche geäußert und Informationen vermittelt[3].

Die Spracherkennung wird als „die maschinelle Umsetzung von gesprochener Sprache in geschriebenen Text"[4] definiert und spielt somit vor allem in der Informatik als zusätzliche Eingabeoption eine wichtige Rolle.

[1] Vgl. Duden (o.J.)
[2] Vgl. TU Berlin (2011)
[3] Vgl. ebenda
[4] Vgl. Pscheidt, S. (o.J.)

Spezifische Aufgaben von Spracherkennungsprogrammen werden im Online-Lexikon für Informationstechnologie benannt: die Analyse der eingegebenen Sprachinformationen im Bezug auf die gesprochenen Wörter, auf deren Bedeutung und hinsichtlich der charakteristischen Merkmale des Sprechers[5]. Sprache wird digitalisiert und anschließend mit vorgegebenen Mustern verglichen, um das Gesprochene als Text darzustellen[6].

Man unterscheidet zwischen diskreter und kontinuierlicher Sprache. Diskrete Sprache beinhaltet Pausen zwischen den Wörtern, sodass diese einzeln erkannt und analysiert werden können. Kontinuierliche Sprache hingegen spiegelt den normalen Redefluss wider[7].

2.2. Geschichtlicher Hintergrund

Die ersten Entwicklungen in Richtung Spracherkennung datieren mehr als 50 Jahre vor der Entwicklung des ersten Computers: Um seiner tauben Frau zu helfen, experimentierte Alexander Graham Bell mit der Sichtbarmachung von Sprache durch den Bau eines Gerätes, das Sprache in Bilder umwandeln sollte; obwohl er hiermit keinen Erfolg hatte, entstand aus diesem Ansatz die Erfindung des Telefons. Die erforderliche Computerleistung zur Entwicklung und privaten Nutzung von Spracherkennungssoftware ist erst seit den 1990ern verfügbar.[8]

Erste moderne Entwicklungen zur Spracherkennungssoftware konnten jedoch nur mit diskreter Sprache arbeiten. Dies war für den Anwender vielmehr hinderlich als hilfreich und fand lediglich Akzeptanz bei Menschen mit Behinderung, die die Tastatur schwerlich oder gar nicht nutzen konnten.[9]

Für den durchschnittlichen Endbenutzer beinhaltete die Nutzung von Spracherkennungssoftware aber zu viele Limitationen: der Bedarf zusätzlicher Hardware (z.B. gute Soundkarte und Mikrofon) erwies sich als umständlich, spezielle Trainings für den Umgang mit der Software waren nötig, und auch finanziell gestaltete sich die

[5] Vgl. IT Wissen (o.J.b)
[6] Vgl. Business Dictionary (o.J.)
[7] Vgl. ebenda
[8] Vgl. Grabianowski, E. (o.J.e)
[9] Vgl. Zumalt, J. R. (2005)

Umstellung schwierig, da das Angebot nur von einigen wenigen Anbietern gedeckt wurde.[10]

Auch Lamont Wood, langjähriger Journalist von Computer World, schreibt, dass bis vor einem Jahrzehnt der Nutzwert von Spracherkennungssoftware nicht die damit einhergehenden Limitationen ausgleichen konnte – Befehle und Eingaben wurden gerade bei kurzen Wörtern oft falsch interpretiert und ausgegeben.[11] Mittlerweile ist die Fehlerrate bei Spracheingaben jedoch trivial[12] – viele der Softwares „lernen" mit dem häufigen Gebrauch der Software[13] und erkennen sowohl die Stimme der Person, die Befehle eingibt, als auch das Gesagte sehr genau[14]. Die neuesten Forschungen werden *Star Trek* - Zukunftsszenarien immer ähnlicher: die Defense Advanced Research Projects Agency (DARPA) setzt zur Zeit drei Forscherteams zur Entwicklung von Global Autonomous Language Exploitation (GALE) ein. Dieses Programm soll fremdsprachige Nachrichten einlesen und direkt in die gewünschte Sprache übersetzen. Als Ergebnis soll eine Sofort-Übersetzung mit einer Genauigkeit von mindestens 90% zustande kommen. Dieselbe Technologie soll auch im Militär eingesetzt werden, um es Soldaten zu ermöglichen mit anders-sprachigen Zivilisten zu kommunizieren.[15]

Seit jeher als vielversprechende Technik der Zukunft angesehen[16] besteht die Möglichkeit, dass sich aus Spracherkennung ein Sprachverstehen entwickelt, das die Bedeutung dessen, was gesagt wird, verstehen und darauf antworten kann. Forscher gehen sogar so weit zu behaupten, dass die Spracherkennung die direkteste Verbindung zwischen moderner Computertechnologie und künstlicher Intelligenz sei[17].

2.3 Technische Grundlagen

Um das gesprochene Wort in Text oder einen für einen Computer verständlichen Befehl umzuwandeln, ist die Ausführung einiger Schritte notwendig: Zunächst tastet ein „Analog/Digital-Wandler"[18] analoge Signale ab, um sie daraufhin zu quantisieren,

[10] Vgl. ebenda
[11] Vgl. Bednar, J. (2014)
[12] Vgl. ebenda
[13] Vgl. BBC (2009)
[14] Vgl. Bednar, J. (2014)
[15] Vgl. Grabianowski, E. (o.J.e)
[16] Vgl. Zumalt, J. R. (2005)
[17] Vgl. Grabianowski, E. (o.J.e)
[18] Vgl. Jahnke, S. (2006), S. 3

codieren und schließlich in digitale Daten umwandeln zu können, die ein Computer verarbeiten kann.[19] Der Wandler digitalisiert den Ton, indem präzise Messungen der Wellen in kurzen Abständen vorgenommen werden. Das System filtert daraufhin den digitalisierten Ton, um ungewollte Störgeräusche zu eliminieren.[20] Nach diesem Schritt wird der Ton zusätzlich in verschiedene Frequenzbänder aufgeteilt. Des Weiteren wird der Ton normalisiert, d.h. in eine konstante Lautstärke umgewandelt[21], und die Geschwindigkeit des Gesprochenen an die Geschwindigkeit der im System gespeicherten Ton-Vorlagen angepasst. Das Signal wird nun in kleinste Segmente von hundertstel- respektive tausendstel-Sekunden aufgeteilt; diese Segmente werden mit bekannten Phonemen der entsprechenden Sprache abgeglichen.[22] Phoneme sind die kleinsten Elemente einer Sprache, von denen es in der deutschen Sprache vierzig gibt.[23]

Hierauf folgen nun die komplexesten zwei Schritte, auf die sich die Forschung im Bereich Spracherkennung vornehmlich fokussiert: Das Programm untersucht die einzelnen aus den vorherigen Schritten ermittelten Phoneme im Kontext der Phoneme, die sich um sie herum befinden. Danach lässt das Programm dieses Gebilde durch ein komplexes statistisches Modell laufen und gleicht es mit einer großen Bibliothek bekannter Wörter, Ausdrücke und Sätze ab. Auf diesem Abgleich basierend bestimmt das Programm nun, was mit größter Wahrscheinlichkeit tatsächlich gesagt wurde und gibt dieses in Textform oder als Befehl aus.[24]
Die letzten beiden Schritte werden im Folgenden näher beschrieben: Frühere Spracherkennungssysteme versuchten, grammatikalische und syntaktische Regeln auf Sprache anzuwenden. Wenn das Gesprochene diesen Regeln entsprach, konnten die Systeme feststellen, was gesagt wurde; allerdings ist Sprache sehr vielseitig - es gibt verschiedene Dialekte, Akzente und Eigenheiten, die das Gesprochene beeinflussen. Diese Variationen konnten von derartigen Spracherkennungssystemen nicht erkannt werden; auch kontinuierliche Sprache ohne Pausen zwischen jedem gesprochenen Wort konnten diese Systeme nicht verarbeiten. Heutzutage nutzen Spracherkennungssysteme, wie oben erwähnt, statistische Systeme, die

[19] Vgl. ebenda, S. 3
[20] Vgl. Grabianowski, E. (o.J.d)
[21] Vgl. Bruns/Ackermann (2005), S. 205
[22] Vgl. Grabianowski, E. (o.J.d)
[23] Vgl. Bruns/Ackermann (2005), S. 196
[24] Vgl. Henning (2007), S. 142

Wahrscheinlichkeiten und mathematische Funktionen heranziehen, um das am ehesten dem zuvor Gesprochenen entsprechende Ergebnis zu liefern.[25] Die dominierenden statistischen Modelle in diesem Feld sind das „Hidden Markov Model“[26] sowie „neural networks“[27], wobei Ersteres das allgemein geläufigere Modell darstellt[28], weshalb nachfolgend näher darauf eingegangen wird:
Beim „Hidden Markov Model“[29] ist jedes Phonem wie das Bindeglied einer Kette zu verstehen: die vollendete Kette bildet das Wort. Das Spracherkennungsprogramm gleicht den digitalisierten Ton Stück für Stück mit den eingespeicherten Phonemen ab. Hierbei wird immer zuerst das Phonem abgeglichen, das am wahrscheinlichsten auf das zuvor ermittelte folgt. Das Programm vergibt bei jedem Phonem Wahrscheinlichkeiten für alle Phoneme, die darauf folgen könnten; die ermittelten Wahrscheinlichkeiten basieren auf dem dem System zugrunde liegenden Wörterbuch und dem Training des Systems. Hat das System das dem digitalisierten Ton entsprechende Phonem ermittelt, so wird das nächste Stück abgeglichen, bis das volle Wort durchlaufen wurde.[30]

Für jedes tatsächlich gesprochene Wort wird hierdurch allerdings nicht nur das **eine** am wahrscheinlichsten zutreffende Wort bestimmt, sondern **mehrere** mit dem tatsächlich Gesprochenen am ehesten übereinstimmende Worte ermittelt. Im Zuge dessen ist noch ein letzter Schritt notwendig, um die am wahrscheinlichsten zutreffende Wortkette auszumachen, zu deren Bestimmung ein Sprachmodell herangezogen wird. Solch ein Sprachmodell kann entweder Grammatiken oder “N-Gram-Modelle”[31] umfassen. Grammatiken werden zur Spracherkennung nur dort eingesetzt, wo die Erkennung bloß auf einem minimalen Wortschatz basiert. Für einen größeren Wortschatz und kontinuierliche Spracherkennung sind die auf Wortwahrscheinlichkeiten beruhenden “N-Gram-Modelle” anzuwenden, mithilfe derer ermittelt wird, mit welcher Wahrscheinlichkeit ein bestimmtes Wort auf vorangegangene Worte folgt.[32] Die vorherigen Worte und das folgende Wort werden zusammen genommen als N-Gramm bezeichnet: besteht der vorangegangene Teil aus nur einem

[25] Vgl. Grabianowski, E. (o.J.b)
[26] Vgl. Henning (2007), S. 142
[27] Vgl. Grabianowski, E. (o.J.b)
[28] Vgl. Euler (2006), S. 67
[29] Vgl. Henning (2007), S. 142
[30] Vgl. Grabianowski, E. (o.J.b)
[31] Vgl. Germesin, S. (2006)
[32] Vgl. ebenda

Wort, so wird das N-Gramm als Bigramm bezeichnet; besteht der vorherige Part aus zwei Worten, so nennt man dieses Wortgebilde Trigramm.[33] Für (Teil-)Sätze muss ein Spracherkennungsprogramm zudem bestimmen, wo einzelne Wörter anfangen und aufhören. Bei einem Vokabular von ungefähr 60.000 Wörtern muss ein Programm somit allein bei drei aneinander gereihten Wörtern unter 216 Billionen möglichen Kombinationsmöglichkeiten wählen, weshalb wie bereits erwähnt ein Training für Spracherkennungsprogramme notwendig ist.

Um Systeme zu trainieren, werden Beispieldaten benötigt, die aus mehreren tausend Stunden gesprochener Sprache und hunderten Megabytes an Text bestehen können. Diese werden dann verwendet, um akustische Modelle sowie Sprachmodelle für Wörter, Wort-Listen und Multi-Wort - Wahrscheinlichkeitsnetzwerke zu erstellen.

Softwareentwickler richten das Anfangsvokabular der Spracherkennungssoftware ein; der Endnutzer übernimmt das weitere Training.[34] Im professionellen Sektor wird die Spracherkennungssoftware bei der ersten Benutzung an den Nutzer angepasst, indem dieser einige Sätze diktiert, um das System an sein spezielles Sprachmuster anzupassen.[35] Auch unternehmensspezifische Begriffe müssen zusätzlich vom Nutzer im System gespeichert werden. Eine Ausnahme stellen Programme dar, die speziell für den medizinischen oder rechtlichen Sektor entwickelt werden.[36]

2.4 Einsatzbereiche von Spracherkennungsprogrammen

Beim Einsatz von Spracherkennungsprogrammen wird zwischen drei Anwendungsbereichen differenziert[37]:

- Gerätesteuerung
- Diktiersysteme
- Sprachdialogsysteme

Spracherkennung kann unter anderem zur **Steuerung von Geräten** eingesetzt werden; durch Spracheingaben wird beispielsweise eine komfortable Bedienung von

[33] Vgl. Reichel, U. (2010)
[34] Vgl. Grabianowski, E. (o.J.b)
[35] Vgl. BBC (2009), S.2
[36] Vgl. Grabianowski, E. (o.J.b)
[37] Vgl. Euler (2006), S.17

Maschinen ermöglicht. Ein häufiger Anwendungsbereich der sprachgesteuerten Gerätebedienung sind Freisprecheinrichtungen im Auto, mithilfe derer nicht nur das Abhalten von Telefonaten, sondern auch die Steuerung weiterer Anwendungen wie der Sprachwahl oder der Bedienung eines Navigationssystems während des Fahrens erheblich erleichtert werden. In Deutschland ist der Einsatz einer KFZ-Freisprecheinrichtung in Personenkraftwagen beim Telefonieren laut § 23 Abs. 1a StVO (D) bei laufendem Motor seit Anfang 2001 gar Pflicht.

Des Weiteren ermöglichen Spracherkennungsprogramme Menschen mit Behinderungen die sprachliche Bedienung von beispielsweise Rollstühlen, Betten oder Telefonen, sowie auch das Arbeiten am Computer ohne Tastatur.[38] Hinsichtlich des zweiten Anwendungsbereiches unterstützen **Diktiersysteme** die automatische Erfassung von Text. Für den Privatgebrauch können derartige Programme dazu genutzt werden, E-Mails oder Texte zu verfassen. Einige Programme wurden für bestimmte professionelle Sektoren wie medizinische Datenübertragungsdienstleistungen entwickelt.[39] Diktiersysteme mit integriertem Fachvokabular eignen sich jedoch nicht nur für Ärzte, sondern auch für Rechtsanwälte, Juristen oder ähnliche Zielgruppen.

Zu guter Letzt wird Spracherkennung für **Sprachdialogsysteme**, auch Voice Portals genannt, eingesetzt. Diese dienen dazu, automatisierte Dienste für Anrufer bereitzustellen.[40] Größere Firmen nutzen die Programme beispielsweise für ihre Service-Hotlines, um bei eingehenden Anrufen erst einmal generell zu klären, was der Anrufende genau möchte. Auch die Auskunftssysteme der Deutschen Bahn oder der Telekom fallen unter die Kategorie der Sprachdialogsysteme.

2.5 Kategorien von Spracherkennungsprogrammen

Die Literatur unterteilt Spracherkennungssoftware anhand von vier Kategorien. Die erste Unterteilung bezieht sich auf die Art der Äußerung, die von den Spracherkennungssystemen erkannt werden kann. Eine weitere Kategorie umfasst die Anpassung der Software an den Sprecher und die Größe des Wortschatzes. Eine letzte Unterteilung bezieht sich auf den Ort, an dem die Verarbeitung durchgeführt wird.

[38] Vgl. ebenda, S.17
[39] Vgl. Grabianowski, E. (o.J.a)
[40] Vgl. Euler (2006), S.17

2.5.1 Art der Äußerung

Einzelworterkenner sind Systeme, die einzelne Wörter oder kurze Kommandos erkennen. Diese müssen isoliert voneinander, ergo mit kurzen Pausen, gesprochen werden.

Keyword-Spotter sind Systeme, die innerhalb einer beliebigen Äußerung einzelne Wörter oder kurze Kommandos erkennen.

Verbundworterkenner sind Systeme, die zusammenhängend gesprochene einzelne Wörter identifizieren können. Das System ist jedoch nicht in der Lage, einen Zusammenhang zwischen den einzelnen Wörtern zu erkennen.[41]

Kontinuierliche Spracherkenner sind Systeme, die fließend zusammenhängend gesprochene Sätze erkennen können. Das System ist zudem in der Lage einen Zusammenhang zwischen den einzelnen Wörtern zu erkennen.[42]

2.5.2 Anpassung an den Sprecher

Sprecherabhängige Systeme werden speziell auf einen einzelnen Sprecher ausgerichtet und müssen daher in einer Trainingsphase an ihn angepasst werden. Dadurch steigert sich die Erkennungsleistung für diesen Sprecher; das System kann hierdurch nur von ihm optimal genutzt werden.[43]

Sprecherunabhängige Systeme sind nicht an einen Sprecher angepasst und erkennen deshalb die Sprache jeder beliebigen anderen Person. Dadurch kann das System von jedem Sprecher benutzt werden, jedoch verschlechtert sich die Erkennungsleistung im Vergleich zum sprecherabhängigen System.[44]

Sprecheradaptive Systeme sind sprecherunabhängige Systeme, die sich aber an einen Sprecher anpassen können. Diese Anpassung findet während einer sehr kurzen Trainingsphase oder während der Benutzung des Systems statt. Beim sprecheradaptiven System werden die Vorteile der sprecherabhängigen und sprecherunabhängigen Systeme vereint. So kann das System von jedem Sprecher benutzt wer-

[41] Vgl. Pfister/Kaufmann (2008), S. 290 ff
[42] Vgl. Bruns/Ackermann (2005), S. 206 ft.
[43] Vgl. Pfister/Kaufmann (2008), S. 292
[44] Vgl. Lee (1989), S. 3ff

den, während sich die Erkennungsleistung speziell für einen Sprecher erhöhen lässt.[45]

2.5.3 Größe des Wortschatzes

Systeme mit ***kleinem Wortschatz*** umfassen maximal einige hundert Wörter. Diese werden meist für Steuerungsaufgaben verwendet. Der Wortschatz ist speziell auf diese Aufgabe zugeschnitten.[46]

Systeme mit ***großem Wortschatz*** umfassen eintausend bis mehrere hunderttausend Wörter. Bei einem Wortschatz dieser Größe kann es aufgrund von phonetisch ähnlichen Wörtern vermehrt zu Verwechslungen kommen. Die Anzahl dieser Verwechslungen steigt signifikant ab einer Anzahl von mehr als 1.000 Wörtern. Daher ist bei Systemen dieser Art ein Sprachmodell notwendig.[47]

2.5.4 Ort der Verarbeitung

Front-End-Systeme sind Systeme, bei denen die Verarbeitung der Sprache und deren Umsetzung in einen Text sofort auf dem Rechner des Sprechers ausgeführt wird. Dadurch kommt es kaum zu einer Zeitverzögerung und der Sprecher benötigt keine drahtlose Verbindung zu einem externen Server.

Back-End-Systeme sind Systeme, bei den die Verarbeitung der Sprache und deren Umsetzung in einen Text auf einem externen Server durchgeführt wird. Dies führt zu einer merklichen Zeitverzögerung und eine drahtlose Verbindung zum Server wird benötigt.[48]

2.6 Schwächen von Spracherkennungssystemen

Spracherkennungssysteme funktionieren auch heutzutage nicht fehlerfrei - sie haben einige Schwächen, zwischen denen zu differenzieren ist. Probleme können bei der

[45] Vgl. Euler (2006), S. 19
[46] Vgl. Pfister/Kaufmann (2008), S. 292
[47] Vgl. Lee (1989), S. 8
[48] Vgl. Wikipedia (2014d)

Spracherkennung immer dann auftreten, wenn das System den Sprecher nicht „versteht“.

Hierfür gibt es eine Reihe von Gründen: Zum einen kann es passieren, dass sich die Sprache mit zu vielen Störgeräuschen, beispielsweise bei der Nutzung in einem Café, vermischt. Zum anderen stellen für Spracherkennungssysteme lokale Dialekte oder Akzente von Nicht-Muttersprachlern eine Herausforderung dar. Hier reichen teilweise schon Unterschiede im Klangbild der Sprache, um das System aus dem Konzept zu bringen. Die meisten dieser Fehler treten vermehrt bei kontinuierlicher Sprache auf und können durch Nutzung diskreter Sprache größtenteils vermieden werden.[49,50] Dabei kann es allerdings auch passieren, dass ein gesprochenes Wort nicht im Wortschatz des Systems vorhanden ist und das System dieses nicht korrekt abbilden kann. Ähnliches gilt, wenn Wörter gleich klingen (vgl. „mehr“ und Meer“).

Auch muss ein Spracherkennungssystem dazu fähig sein, Füllwörter herauszufiltern, die vom Sprecher zwar gesagt werden, aber beim Diktieren nicht in den Text gehören; dies gelingt allerdings sehr selten.[51] Die gleichzeitige Erkennung unterschiedlicher Stimmen, um beispielsweise ein Meeting schriftlich zu dokumentieren, funktioniert mit aktuellen Lösungen wenig bis gar nicht.[52]

2.7 Auswahlmethoden

Im Folgenden werden unter 2.7.1. die Gesichtspunkte, die sich in Absprache mit der .Versicherung als Auftraggeber ergeben haben, theoretisch abgehandelt. Darauffolgend werden unter 2.7.2. die Kriterien herausgearbeitet, die vor der Installation eines Softwaresystems getestet werden können. Anschließend wird unter 2.7.3. auf die Gewichtung dieser Aspekte eingegangen.

Zum Schluss werden unter 2.7.4 die Merkmale beschrieben, die erst nach der Installation beurteilt werden können. In Ergänzung zu den folgenden Abschnitten werden ein Anforderungsprofil (zu 2.7.1.), ein Kriterienkatalog mit Gewichtung (zu 2.7.2. und dessen Unterpunkte sowie 2.7.3.) sowie ein Testkatalog (zu 2.7.4.) erstellt (siehe Anhang).

[49] Vgl. Pfister/Kaufmann (2008), S. 286 ff.
[50] Vgl. Lee (1989), S. 12 f.
[51] Vgl. Pfister/Kaufmann (2008), S. 286 ff.
[52] Vgl. Grabianowski, E. (o.J.c)

2.7.1. Anforderungsprofil

Softwarelizenz

Eine Softwarelizenz regelt die Bedingungen für die Nutzung oder Weitergabe der Software.[53] Es gibt verschiedene Lizenzarten: Bei **kommerzieller Software** muss kostenpflichtig ein Nutzungsrecht erworben werden. Die Software ist nur für den persönlichen Gebrauch bestimmt.

Bei **Shareware** hingegen kann die Software kostenlos für einen bestimmten Zeitraum getestet sowie beliebig vervielfältigt werden. Hierbei wird jedoch kein Zugriff auf den Quellcode gewährt. Nach Ablauf des Testzeitraums muss die Lizenz zur Nutzung erworben werden.

Freeware ist Software, die kostenlos und unbefristet genutzt werden kann. Dabei gibt der Urheber den Quellcode jedoch nicht frei.[54]

Open Source - Software wird vom Urheber kostenlos und unbefristet zur Verfügung gestellt. Zusätzlich ist der Quellcode einsehbar und darf weiterverarbeitet werden.

Die **Public Domain** - Lizenz ist in Deutschland nicht zulässig, da der Urheber bei dieser Lizenz alle seine Urheberrechte abtritt, wodurch die Software uneingeschränkt genutzt und weiterverwendet werden könnte. Auch wenn in Deutschland diese Art von Lizenz unzulässig ist, kann Public Domain Software in anderen Ländern genutzt werden.[55]
Die Vorgaben der .Versicherung als Auftraggeber dieses Projekts verlangen eine Open Source - Software-Lösung. Daher wird dieses Kriterium als Teil des Anforderungsprofils gewertet. Spracherkennungssoftware, die keine Open Source - Lizenz aufweist, wurde deshalb nicht in Betracht gezogen.

Ort der Verarbeitung

Die Verarbeitung der Sprache kann zum einem direkt am Gerät des Benutzer und zum anderen auf einem externen Server erfolgen. Wird die Verarbeitung auf dem

[53] Vgl. Jura Forum (2011)
[54] Vgl. Landesakademie für Fortbildung und Personalentwicklung an Schulen (o.J.)
[55] Vgl. Universität Oldenburg (o.J.)

Gerät des Benutzers durchgeführt, spricht man von einem Front-End-System. Im Gegensatz dazu findet bei Back-End-Systemen die Verarbeitung auf einem externen Server statt.[56]

Der Datenschutz ist ein wichtiges Thema bei der Auswahl der Software. Da die Spracherkennung später Teil einer Applikation sein soll, die Kundendaten enthält, muss verhindert werden, dass diese sensiblen Informationen auf einem fremden Server verarbeitet werden. Daher wird dieses Kriterium als Teil des allgemeinen Anforderungsprofils bewertet. Es wird nur Spracherkennungssoftware in Betracht gezogen, die sich entweder als Front-End-Lösung oder aber als Back-End-Lösung mit einem eigenen Server umsetzen lässt.

Spracherkennungstypen

Wie in 2.4 bereits angeführt, gibt es für Spracherkennungssoftware drei Einsatzgebiete: Gerätesteuerung, Diktiersysteme und Sprachdialogsysteme. Da die Spracherkennungssoftware später in eine Applikation eingebunden werden soll, um mündliche Schadensberichte in Text umzuwandeln, muss die Software die Diktiersystem- Funktion erfüllen. Daher wird dieses Kriterium als Teil des Anforderungsprofils definiert und lediglich Spracherkennungssoftware in Betracht gezogen, die mindestens die Diktierfunktion erfüllt, ergo kontinuierliche Sprache in Text umwandeln kann.

Sprache

Nicht jede Spracherkennungssoftware ist in allen Sprachen erhältlich. Da die Software benutzt werden soll, um deutsche Sprache in Text umzuwandeln, bildet die Existenz eines umfassenden deutschen Vokabulars eine Grundvorrausetzung und ist damit dem Anforderungsprofil zuzuordnen. Es wird nur Software in Erwägung gezogen, die die deutsche Sprache unterstützt.

2.7.2. Kriterien vor Installation

Die Kriterien vor der Installation stellen alle Kriterien dar, die ohne Installation und

[56] Vgl. IT Wissen (o.J.a)

Test der Software, sondern durch Recherche oder Informationen des Herstellers zum Vergleich herangezogen werden können.

2.7.2.1 Genauigkeit

Zum Thema Genauigkeit werden im Folgenden

- die Größe des Wortschatzes
- das Sprachmodell
- die Sprecherabhängigkeit
- die Sprachart
- die Erkennungsgenauigkeit

als Kriterien behandelt.

Wortschatz

Der Wortschatz einer Spracherkennungssoftware ist essentiell um zu beurteilen, wie gut die Software im täglichen Gebrauch bestehen kann. Der Wortschatz sollte unabhängig vom Sprecher sein, d.h. es werden nur die Referenzmuster (die Aussprechart) an den Sprecher angepasst, nicht aber der eigentliche Wortschatz[57], da die Benutzer der Software variieren können.

Da die Software nicht nur zur Steuerung bestimmter Befehle, sondern zur sprachlichen Eingabe ganzer Textpassagen verwendet werden soll, wird ein großer Wortschatz mit einem Sprachmodell benötigt.[58] Umso größer der Wortschatz ist, desto flexibler ist das Programm in der Worterkennung, da es das Gesagte mit mehr Beispielwörtern abgleichen kann. Während in den Anfängen der Spracherkennung schon ein Wortschatz von 1.000 Wörtern als Durchbruch galt[59], sind heute mehr als 150.000 Wörter keine Seltenheit.[60]

Um eine Vergleichbarkeit herzustellen, wird die Größe des Wortschatzes betrachtet; je mehr Wörter von der Software in Kombination mit einem geeigneten Sprachmodell beherrscht werden, desto mehr steigert sich die Genauigkeit der Spracherkennung und desto besser fällt die Punktebewertung anhand des Kriterienkataloges aus.

[57] Vgl. FH Wedel (o.J.)
[58] Vgl. 2.5.3. „Größe des Wortschatzes“
[59] Vgl. Computerwoche (2012)
[60] Vgl. Nuance (2011)

Sprachmodell

Beim Sprachmodell wird zwischen den verschiedenen Arten der eingesetzten Statistik unterschieden. Diese Statistik hilft vor allem bei längeren Spracheingaben die Wahrscheinlichkeit aufeinanderfolgender Wörter abzugleichen und somit ein korrektes Ergebnis zu liefern[61]. Angewandt werden hierbei verschiedene N-Gramme. Üblich sind Bi-, Tri-, Tetra- oder Quintgramm-Statistiken. Je höher „n" ist, desto mehr Wörter werden im Kontext ihrer Wahrscheinlichkeit abgeglichen und desto genauer ist die Spracherkennung.[62]

Daher werden in der Kriterienbewertung z.B. Trigramme höher gewertet als Bigramme.

Sprachabhängigkeit

Bei der Bewertung der Sprachabhängigkeit wird betrachtet, ob es sich um ein sprecherabhängiges, sprecherunabhängiges oder sprecheradaptives System handelt. Im Gegensatz zum sprecherabhängigen System können sprecherunabhängige und sprecheradaptive Systeme vonmehreren Sprechern benutzt werden (siehe auch 2.5.2).[63]

Da anhand der vorgegebenen Problemstellung nicht sichergestellt werden kann, dass die Sprecher nicht wechseln, wird sprecherunabhängige und –adaptive Open Source-Software positiv gewertet, während ein sprecherunabhängiges System eine negative Bewertung erhält.

Sprachart

Die Sprachart besagt, wie Befehle für die Spracherkennungssoftware eingegeben werden müssen, damit das System diese erkennt.

Unterschieden wird zwischen Einzelworterkennern (diskrete Spracherkennung), Keyword-Spottern, Verbundworterkennern und kontinuierlichen Spracherkennern[64] (siehe auch 2.5.1). Die Unterscheidung in diese verschiedenen Spracharten ist wichtig, um die Genauigkeit einer Spracherkennung weiter zu spezifizieren. So sind die Anforderungen an die Genauigkeit einer diskreten Spracherkennungssoftware gerin-

[61] Vgl. dc-Systeme Informatik GmbH (2012)
[62] Vgl. Universität Bonn (o.J.)
[63] Vgl. Klaus Fellbaum (2012)
[64] Vgl. 2.5.1 „Art der Äußerung"

ger als bei einem kontinuierlichen System, da bei diesen zusätzliche Sprachmodelle zur Analyse benötigt werden.

Darüber hinaus ist für den Einsatz der Software im Vertrieb nur die kontinuierliche Spracherkennung von Interesse, da die diskrete Spracherkennung für lange Spracheingaben nicht geeignet ist[65] und somit dem Ziel der Effizienzsteigerung des Vertriebs hinderlich wäre.

2.7.2.2 Leistungsanforderungen

In Bezug auf die Leistungsanforderungen einer Spracherkennungssoftware werden folgende Punkte untersucht: das Betriebssystem und die Integrationsfähigkeit der Software im Hinblick auf die Nutzung von APIs oder Webservices.

Betriebssystem

Die verschiedenen Spracherkennungslösungen setzen unterschiedliche Betriebssysteme voraus. Die gängigen Betriebssysteme für Tablets sind iOS, Android und Windows.[66] Da vom Auftraggeber momentan iPads mit iOS Betriebssystem verwendet werden, wäre es von Vorteil, wenn die ausgewählte Software auf diesem System liefe. Android und Windows sind zwar auch denkbar, allerdings mit einem gewissen Mehraufwand verbunden. Daher erhalten Spracherkennungslösungen, die mit allen Betriebssystemen kompatibel sind, fünf Punkte. Lösungen, die mindestens für iOS verfügbar sind, erhalten drei Punkte. Dagegen werden Lösungen, die mit Android oder Windows, oder beidem vereinbar sind, mit einem Punkt gewertet.

Integrationsfähigkeit

Damit die Spracherkennungssoftware innerhalb der .Versicherung-Applikation genutzt werden kann, muss sie in das bereits bestehende Programm integriert werden. Dies ist gegebenenfalls sehr aufwendig, da es deshalb einiger Modifikationen des Codes bedarf. Daher ist es von Vorteil, wenn sich die Software hier gut integrieren lässt.[67]

Hierfür gibt es mehrere Möglichkeiten:

[65] Vgl. Linguatec (o.J.)
[66] Vgl. Rupp, M. (2012)
[67] Vgl. PC Mag (o.J.)

Eine Möglichkeit ist die Nutzung eines Application-Programming-Interface (API), das eine Zusammenstellung von Anleitungen und Standards bietet, um eine webbasierte Applikation oder ein Web-Tool anzusprechen. Ein API ermöglicht die Kommunikation zwischen verschiedenen Software-Systemen. APIs werden von der jeweiligen Herstellerfirma veröffentlicht, um den Zugang zu ihrem System und dessen Nutzung zu ermöglichen.[68]

Eine weitere Möglichkeit ist die Verwendung eines Webservices. Webservices erlauben ebenfalls eine Kommunikation zwischen verschiedenen Softwaresystemen. Allerdings muss dies per Intra- oder Internet geschehen. Die Client-Applikation sendet eine Anfrage für eine integrierte Applikation zu einem Webserver. Dort wird diese Anfrage ausgeführt und das Ergebnis zurückgeschickt. Da die Anfrage in URLs, HTTP und XML stattfindet, können Applikationen auf jeder Plattform einen solchen Webservice verwenden.[69] Um die Software später einbinden zu können, ist eine Integration essentiell. Dabei spielt es keine Rolle, welche der beiden Methoden verwendet werden. Daher wird eine Software, die sowohl APIs als auch Webservices einbinden kann, mit fünf Punkten bewertet, während eine Software, die lediglich eine Integrationsmöglichkeit bietet, drei Punkte erhält. Besteht gar keine Möglichkeit zur Einbindung, werden keinerlei Punkte vergeben.

2.7.2.3 Zusatzfunktionen

Hinsichtlich zusätzlicher Funktionen von Spracherkennungssoftware werden die Möglichkeit zur Audio-Aufnahme, die Erfordernis eines externen Mikrofons, die Unterstützung von Bluetooth-Geräten, der Import und Export von Wörterbüchern, sowie die Sicherung von Nutzerprofilen untersucht.

Audio-Aufnahme

Manche Spracherkennungssoftware bietet zusätzlich zur automatischen Verschriftlichung von Diktaten eine Funktion zur Audio-Aufnahme. Dies kann sich als sehr nützlich erweisen, falls die eingegebenen Texte von der Software nicht korrekt verschriftlicht wurden und Unklarheit über den Inhalt des tatsächlich Gesprochenen herrscht.

[68] Vgl. Roos, D. (o.J.)
[69] Vgl. Samtani, G. (2002)

Hierdurch können der diktierte Text noch einmal abgehört und etwaige Fehler korrigiert werden.

Externes Mikrofon erforderlich

Am Markt werden verschiedene Eingabegeräte für Spracherkennungsprogramme angeboten. Zunächst eignen sich professionelle Diktiergeräte für hochwertige Audio-Aufnahmen, wohingegen Handmikrofone eine kostengünstigere Alternative darstellen. Auch Drahtlos-Headsets erfreuen sich großer Beliebtheit, da sie eine komfortable Bedienung und Handhabung ermöglichen.

Eine wichtige Rolle spielt hierbei die Genauigkeit bei der Audio-Aufnahme. Experten vertreten die Ansicht, dass die Erkennungsleistung erheblich höher sei, wenn auf einen stets konstanten Abstand zwischen Mikrofon und Mund geachtet werde.[70] Aus diesem Grund gelten Desktop-Mikrofone als eher ungeeignet, da sie abhängig von der Entfernung und des Winkels zum Sprecher eine höhere Störanfälligkeit aufweisen.[71]

Bluetooth-Unterstützung

Wie bereits erwähnt, sind gegebenenfalls spezielle Eingabegeräte für die Spracheingabe am Computer oder auf mobilen Geräten erforderlich. Bei Drahtlos-Headsets wird zwischen zwei verschiedenen Techniken unterschieden: Zum einen kann das sogenannte Dect-Protokoll, das auch bei schnurlosen Telefonen eingesetzt wird, verwendet werden. Anbieter sind beispielsweise Jabra, Planatronics oder Sennheiser, jedoch kann es bei einigen Geräten zu Problemen bei der Verwendung am Computer kommen. Deshalb besteht die Möglichkeit statt des Dect-Protokolls ein Standardprotokoll von Bluetooth einzusetzen. Da dieses jedoch nicht unerhebliche Einbußen in der Akustik aufweist, wurden spezielle Bluetooth-Protokolle entwickelt, um diesem Problem entgegenzuwirken. Diese weisen einen verbesserten breitbandigen Audio-Codec auf und sollen die Qualität, Kontinuität und Verlässlichkeit der Audiosignale verbessern.[72]

Wörterbuch Import/Export (Fachvokabular)

[70] Vgl. FAZ (2013)
[71] Vgl. SpeechRecognition Solutions (o.J.)
[72] Vgl. SpeechRecognition Solutions (o.J.)

Ein weiteres Kriterium stellt die Möglichkeit der Erweiterung des Wortschatzes einer Spracherkennungssoftware dar. Zum Teil lässt sich dieser um ein spezifisches Fachvokabular erweitern. Somit können gesprochene Texte bei der Aufnahme mit einem gespeicherten Wörterbuch verglichen werden.[73]

Sicherung von Nutzerprofilen

Im Kriterienkatalog wird auch die Verfügbarkeit und Sicherung von Nutzerprofilen berücksichtigt. Bei der Spracherkennung werden die zum Erkennen der Stimme der jeweiligen Person benötigten Informationen in Nutzerprofilen gespeichert. Bei manchen Spracherkennungsprogrammen besteht auch die Möglichkeit, zusätzliche Profile für ein und dieselbe Person zu erstellen. Der Einsatz unterschiedlicher Profile kann zur Anpassung an die jeweilige Umgebung und deren Geräuschpegel dienen und die Verwendung verschiedener Mikrofone unterstützen.[74]

Je genauer eine Spracherkennungssoftware mithilfe akustischer Trainings auf die Sprechweise eines Benutzers und sein jeweiliges Einsatzgebiet angepasst ist, desto bessere Ergebnisse liefert sie.[75]

*2.7.*2.4 Support

Wichtig für eine Firma wie der .Versicherung ist des Weiteren der Support einer genutzten Software in Form eines Callcenters, Helpdesks o.Ä., da eventuelle Fehlfunktionen dieser dazu führen, dass die Software nicht in vollem Umfang verwendet werden kann. Dies kann gegebenenfalls zu Störungen im Betrieb führen und finanzielle Schäden verursachen.

Dokumentation

Im Hinblick auf die Dokumentation einer Software ist vor allem die Benutzerdokumentation von großer Relevanz. Diese Art der Dokumentation soll dem Anwender einer Software die Nutzung dieser erleichtern und darüber hinaus als allgemeines Nachschlagewerk dienen. Um diesen Zweck zu erfüllen, ist es primär wichtig, dass die Dokumentation eine Bedienungsanleitung beinhaltet. Auch ein Glossar sowie Ansätze zur Lösung bei häufig auftretenden Problemen oder Fragen können für den

[73] Vgl. Dworatschek (1989), S.295
[74] Vgl. Microsoft (o.J.)
[75] Vgl. PC Welt (o.J.)

Anwender hilfreich sein. Je nach Produkt sollten auch etwaige Schnittstellen sowie deren Einbindung ausreichend dokumentiert sein.[76]

Support Community/Support Helpdesk

Wie eingangs erwähnt, können Fehlfunktionen einer Software Unternehmen sowohl finanziell beeinträchtigen als auch zusätzlich deren Ruf schaden. Bei derartigen Problemen können ein Helpdesk sowie die Support Community für eine Software die nötige Abhilfe schaffen. Darüber hinaus tauschen sich auf verschiedenen Portalen im Netz Nutzer über Probleme einer Anwendung aus und unterstützen sich gegenseitig. Ein Helpdesk vonseiten des Entwicklerteams ist wünschenswert, allerdings keine zwingende Voraussetzung. [77,78]

Aktive Weiterentwicklung

Wie auch bei kommerzieller Software ist es für Open Source - Produkte wichtig, dass sie kontinuierlich weiterentwickelt werden, um externen Gegebenheiten gerecht zu werden und durch neue Funktionen Wettbewerbsvorteile zu bieten. Die aktive Weiterentwicklung einer Software kann anhand der Existenz aktiver Produkt-Foren - sogenannter Issue-Tracker, in denen Fehler in der Software aufgenommen und abgearbeitet werden - sowie anhand der Zyklen, in denen neue Software-Versionen veröffentlicht werden, überprüft werden.[79]

2.7.3. Gewichtung der Kriterien

In Abstimmung und nach den Vorgaben der .Versicherung wurde eine Gewichtung der oben genannten Kriterien unter den Abschnitten 2.7.2.1. bis 2.7.2.4. festgelegt; die Punktevergabe von 1 bis 5 spiegelt dabei die Relevanz der Kriterien für den Auftraggeber wider. Eine 5 bedeutet, dass ein Kriterium als sehr wichtig eingestuft wird. Eine 1 bedeutet, dass ein Aspekt nur geringfügig relevant ist.

[76] Vgl. Lehner (1994), S. 14 ff
[77] Vgl. Heise Open Source (o.J.)
[78] Vgl. Noyes, K. (o.J.)
[79] Vgl. Wheeler, D. A. (2011)

Die Gewichtung entsteht somit durch eine Gegenüberstellung der einzelnen Kriterien, wodurch eine Rangfolge bezüglich der Bedeutung der Kriterien festgelegt werden kann[80].

Diese Wertigkeitsfaktoren werden als Multiplikator mit den später in der Analyse zu vergebenden Punkten für die einzelnen Softwareanwendungen verrechnet. Auf diese Weise soll zu einem Gesamtergebnis pro Kriterium gelangt werden, das die Wichtigkeit jenes Kriteriums miteinbezieht.

2.7.4. Kriterien nach Installation

Im Folgenden werden die Kriterien nach der Installation theoretisch beschrieben. Ergänzend zu der schriftlichen Beschreibung der Kriterien wird ein Testkatalog erstellt. Anhand dieses Testkataloges (siehe Anhang 3) kann die Fehlerhäufikeit, Dauer der Verarbeitung, CPU-Last und RAM-Nutzung der beiden Softwarelösungen Pocket-Sphinx und Dragon NaturalSpeaking jeweils mit und ohne Lärmbelastung anhand von 5 Testpersonen abgelesen werden.

Erkennungsgenauigkeit

Unter dem Kriterium Erkennungsgenauigkeit wird überprüft, wie hoch die Genauigkeit einer Software bei der Anwendung ist.

Hierbei werden Prozentangaben, die auf Tests anderer Anwender basieren, zum Vergleich herangezogen. Je genauer die Software im Test abgeschnitten hat, d.h. je mehr Wörter korrekt erkannt wurden, desto mehr Punkte erhält die Software bei der Auswertung.

Verarbeitungsdauer

Die Dauer der Umwandlung einer Sprachnachricht in Text hängt von vielen Faktoren ab:

In erster Linie spielt hierbei die Leistungsstärke des Prozessors des eingesetzten Geräts eine wichtige Rolle[81], allerdings können sich Spracherkennungssysteme selbst bei ein- und demselben Gerät in ihrer Verarbeitungsdauer deutlich voneinander unterscheiden.

[80] Vgl. University Dortmund (o.J.)
[81] Vgl. SpeechRecognition Solutions (2013)

Faktoren, die auf diese Verarbeitungsdauer Einfluss nehmen, sind sowohl die Größe des Vokabulars der Spracherkennungssoftware als auch die Komplexität des von dem Programm verwendeten Sprachmodells. Dabei gilt die Faustregel: Je komplexer und genauer das System arbeitet, desto langsamer ist es.[82]

Ein weiterer Faktor ist zudem, ob die Verarbeitung direkt auf dem Gerät oder auf einem externen Rechner durchgeführt wird.[83]

Die Dauer der Verarbeitung wird als Kriterium für den Testprozess definiert, da eine zeitnahe Verarbeitung der Sprache für den alltäglichen Gebrauch, vor allem im Außendienst einer Versicherung wie der .Versicherung, eine elementar wichtige Eigenschaft darstellt. Die Systeme werden im Testprozess demnach unter Anderem anhand ihrer Verarbeitungsdauer verglichen.

CPU-Last

Die CPU-Last wird üblicherweise in Prozent angegeben und bezeichnet die Rechenzeit, die gleichzeitig beansprucht wird, um laufende Prozesse auszuführen.[84] Je weniger CPU-Leistung ein System beansprucht, desto mehr andere Prozesse können parallel bearbeitet werden und desto schneller wird eine Aufgabe ausgeführt. Dieses Kriterium ist von besonderer Bedeutung, da die zu ermittelnde Open Source - Applikation auf einem Tablet verwendet wird. Dort ist die Rechenkraft limitiert und eine effiziente CPU-Benutzung somit besonders wichtig.

RAM-Nutzung

Die RAM-Nutzung zeigt an, wie hoch die Auslastung des Arbeitsspeichers ist, um die notwendigen Daten für die laufenden Prozesse zu laden. Von der RAM-Nutzung hängt ab, wie viele Programme gleichzeitig geladen und wie schnell die entsprechenden Daten verarbeitet werden können.[85] Dieses Kriterium wird berücksichtigt, da die spätere Applikation auf einem Tablet verwendet werden soll. Dort ist die RAM-Größe limitiert und eine geringe RAM-Nutzung von besonderer Wichtigkeit.[86]

[82] Vgl. Lai, S. L. C./Zhao, Q. (o.J.)
[83] Vgl. Mims, C. (2014)
[84] Vgl. Handermann, S. (2013)
[85] Vgl. Appelt, J. (o.J.)
[86] Vgl. Shanklin, W. (2013)

3 Praxis

Im Folgenden wird zunächst anhand eines Prozessschaubildes die praktische Vorgehensweise erläutert, die angewendet wird, um schlussendlich die am besten für die .Versicherung geeignete Software herauszustellen.

Darauffolgend werden die unter 2.7.1. erläuterten Kriterien des Anforderungsprofils auf eine Liste von 22 durch Recherche herausgestellten Open Source - Programmen angewandt. Die direkte Gegenüberstellung ist im Anhang mittels eines Anforderungsprofils visualisiert.

Des Weiteren werden die unter 2.7.2. angeführten Kriterien und deren Gewichtung unter 2.7.3. herangezogen, um die drei verbleibenden Open Source - Software-Lösungen sowie eine kommerzielle Vergleichssoftware unter Abschnitt 3.3. zu analysieren und im Anschluss unter 3.3.1. zu vergleichen. Die unter 2.7.4. aufgeführten Kriterien für den Testprozess werden unter 3.4. und den Unterpunkten dieses Abschnitts angewandt.

3.1. Prozess

Ziel des Projektes ist es, aus allen 22 untersuchten Open Source - Spracherkennungsprogrammen die Lösung herauszustellen, die am besten den Anforderungen der .Versicherung als Auftraggeber und den Gegebenheiten des Unternehmens gerecht wird. Dabei wird folgender Prozess in Abbildung 1 befolgt:

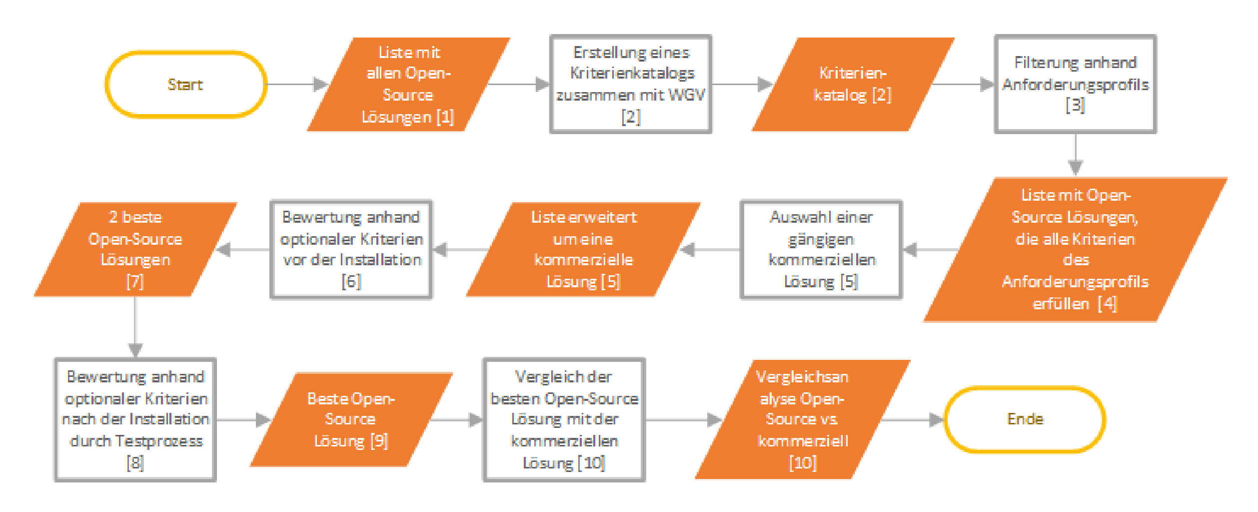


Abb. 1: Flowchart Testprozess

Zu Beginn wird eine Liste mit allen ermittelten Open Source - Spracherkennungsprogrammen erstellt [1]. Im nächsten Schritt wird gemeinsam mit der .Versicherung ein

Anforderungsprofil sowie ein Kriterienkatalog mit entsprechender Gewichtung erstellt. Hierbei werden sowohl die Kriterien festgelegt, deren Prüfung bereits vor der Installation möglich ist, als auch die Kriterien, die erst durch einen Test nach Installation der Systeme getestet werden können. Die Gesichtspunkte des Anforderungsprofils müssen als Mindestvoraussetzung erfüllt werden, die Aspekte des Kriterienkataloges sind optional. Jedes dieser optionalen Kriterien erhält einen Wertigkeitsfaktor, der aussagt, wie wichtig dieses Kriterium im Vergleich zu den anderen ist [2].

Danach wird die unter [1] angefertigte Liste anhand des Anforderungsprofils gefiltert [3]. Übrig bleiben nur jene Programme, die dem Anforderungsprofil gerecht werden [4]. Zu Vergeichszwecken wird für die darauffolgende Prüfung anhand des Kriterienkataloges zudem eine gängige kommerzielle Software ausgewählt [5]. Nun wird diese neue Liste anhand der optionalen Kriterien bewertet, die vor der Installation des Programmes getestet werden können [6]. Anhand der Bewertung der einzelnen Kriterien und deren Wertigkeit werden die beiden Open Source - Lösungen mit den besten Ergebnissen ausgewählt [7]. Diese Programme und das kommerzielle Programm werden installiert und anhand der Kriterien nach der Installation in einem Testprozess getestet [8]. Das Open Source - Programm mit der höchsten Punktzahl wird, basierend auf den zuvor festgelegten Kriterien, als das am besten geeignete Spracherkennungssystem für die Applikation der .Versicherung bewertet [9]. Das Ergebnis dieser Open Source - Lösung wird zudem abschließend mit dem Ergebnis des kommerziellen Programmes verglichen [10].

3.2. Vor-Analyse anhand des Anforderungsprofils

Die Vor-Analyse erfolgt durch die Anwendung des Anforderungsprofils. Dieses beinhaltet vier Mindestvoraussetzungen, die unter dem Abschnitt 2.7.1. genauer erläutert werden:

Die Software soll eine Open Source - Anwendung sein. Der Ort der Verarbeitung der Daten soll entweder als Front-End-Lösung, oder als Back-End-Lösung auf einem eigenen Server umsetzbar sein. Die Software soll die Diktiersystem-Funktion erfüllen und deutsche Sprache erkennen und in Text umwandeln können. Als Ausgangspunkt liegt eine Gesamtliste der 22 durch Recherche ermittelten Open Source - Spracherkennungsprogramme vor (siehe Anhang 4).

Einer der von der .Versicherung als Mindestvoraussetzung vorgegebenen Gesichtspunkte ist, dass die zu ermittelnde Software als Open Source - Anwendung verfügbar sein muss (vgl. 2.7.1. unter dem Oberbegriff *Softwarelizenz*). Dieses Kriterium wird von dem Produkt namens **Gvoice** nicht erfüllt, da es nur käuflich zu erwerben ist.[87] Die Software **RWTH ASR** scheidet aus, da sie nicht für kommerzielle Zwecke genutzt werden darf und demnach nicht für den von der .Versicherung vorgesehenen Gebrauch geeignet ist.[88]

Eine weitere Voraussetzung vonseiten des Auftraggebers ist, dass die gesuchte Software eine Diktiersystem-Funktion enthalten muss (vgl. 2.7.1. unter dem Begriff *Spracherkennungstypen*). Diese Voraussetzung ist bei zehn der 22 ermittelten Programme nicht gegeben:

Die Anwendung **eSpeak** ist eine reine text to speech - Software, d.h. sie ist nur fähig, Text in Sprache umzuwandeln, nicht aber umgekehrt.[89] **HDecode (Hidden Markov Model Toolkit)** hingegen ist ein reines Toolkit und bietet in Folge dessen keine Diktier-Funktion.[90] Mit **Ivona** kann, wie auch bei eSpeak, nur Text in Sprache umgewandelt werden.[91] Die Software **Jialong He's Speech Recognition Research Tool** ist ein Tool zur Forschung, bei dem ebenso keine Spracherkennung möglich ist.[92] Darüber hinaus beinhaltet das **Modular Audio Recognition Framework** keine Spracherkennung, sondern ist eine in Java geschriebene Open Source - Forschungsplattform.[93] **Myers' Hidden Markov Model Software** ist reiner Code in der Sprache C++, der zum einen dazu genutzt werden kann, ein eher experimentelles Spracherkennungssystem zu erstellen und zum anderen dabei behilflich sein kann, die Funktionsweise von Hidden Markov Model - Systemen (vgl. 2.3) nachzuvollziehen. Es ist allerdings keine Spracherkennungssoftware und wird somit den Mindestanforderungen der .Versicherung nicht gerecht.[94] Das **NICO ANN Toolkit** ist ein reines Instrumentarium. Es könnte nur in Kombination mit zusätzlicher Hidden Markov Model - Software zu einer individuellen Spracherkennungssoftware weiterentwickelt

[87] Vgl. Windows (o.J.)
[88] Vgl. Universität Aachen (2014)
[89] Vgl. Sourceforge (o.J.f)
[90] Vgl. htk (o.J.)
[91] Vgl. IVONA Software (o.J.)
[92] Vgl. Universität Ulm (1996)
[93] Vgl. Wikipedia (o.J.)
[94] Vgl. Myers, R. (1996)

werden und erfüllt somit auch nicht den Zweck der Spracherkennung.[95] **Open Mind Speech** ist ein Open Source - Projekt, das nicht vollständig betriebsbereit ist und somit statt zur Nutzung für die Spracherkennung in erster Linie für Entwickler geeignet ist.[96] Die Systeme **Simon** sowie **sphinxkeys** sind beides Programme, die nur als Ersatz für Tastatur und Maus zur Sprachsteuerung von bspw. Computern geeignet sind und aus diesem Grund nicht der Spracherkennung in dem Sinne diesnen, wie es die .Versicherung benötigen würde.[97,98]

Sieben der 22 untersuchten Programme sind nicht fähig, deutsche Sprache in Text umzuwandeln, weshalb die folgenden Systeme ausgeschlossen werden: Das Programm **CVoiceControl/ kVoiceControl** ist nur in englischer Sprache verfügbar.[99] **OpenEars** kann bloß in englischer sowie spanischer Sprache verwendet werden.[100] **Emacspeak** ist eine Anwendung, die ausschließlich auf Englisch verfügbar ist.[101] Das sogenannte **Festival Speech Synthesis System** kann ebenfalls nicht auf Deutsch verwendet werden.[102] Auch **ISIP** erkennt nur englische Sprache.[103] Ein Programm namens **Julius** ist auf die Sprachen Japanisch sowie Englisch ausgerichtet und demnach auch nicht darauf ausgelegt, deutsche Sprache in Text umzuwandeln.[104] Das sechste und somit letzte aufgrund der Sprache ausscheidende System ist **Kaldi**, das nur gesprochenes Englisch in Text transformieren kann.[105]

Durch Recherche anhand der Mindestvoraussetzungen verbleiben somit schlussendlich drei der 22 Programme, nämlich **pocketSphinx**, **Sphinx 4** und **XVoice** zur weiteren Analyse.

3.3. Analysen anhand der Kriterien

Im Folgenden werden die drei verbliebenen Programme sowie ein kommerzielles Vergleichsprodukt anhand der Kriterien analysiert. Durch die anschließende Gegen-

[95] Vgl. KTH (o.J.)
[96] Vgl. TLDP (o.J.)
[97] Vgl. Simon listens (2012)
[98] Vgl. Sphinxkeys (2011)
[99] Vgl. Kiecza (o.J.)
[100] Vgl. Politepix (o.J.)
[101] Vgl. Wikipedia (2014a)
[102] Vgl. Wikipedia (2014b)
[103] Vgl. Sundaram, R. et al. (o.J.)
[104] Vgl. Wikipedia (2014c)
[105] Vgl. Sourceforge (2015)

überstellung der Anwendungen unter Abschnitt 3.3.1., vor allem auch mit der weltweit am häufigsten verkauften Spracherkennungssoftware[106] Dragon NaturallySpeaking, soll ein Vergleich zwischen den Eigenschaften der verfügbaren Open Source - Software und einem kommerziellen Tool hergestellt werden. Diese Analysen der Kriterien werden durch eine übersichtliche Darstellung als Kriterienkatalog im Anhang ergänzt.

Dragon NaturallySpeaking 13

Die Analyse der Software Dragon NaturallySpeaking erfolgt auf Grundlage der aktuellen Version Nummer 13. Zur Vereinfachung und verbesserten Übersichtlichkeit wird im Folgenden auf diese Software lediglich mit dem Namen Dragon verwiesen. Der **Wortschatz** bei Dragon beträgt standardmäßig über 250.000 Wörter[107]. Das **Sprachmodell** kann frei vom Benutzer gewählt werden. Je nach Art des verwendeten Gerätes und der benötigten Genauigkeit sind unterschiedliche Sprachmodelle vorteilhaft. Dragon unterstützt Sprachmodelle mit Trigrammen, Tetragrammen und Quantengrammen[108].

Dragon zählt zu den **sprecheradaptiven** Spracherkennungssoftwaretypen und passt sich demnach an den Sprecher an.[109]

Des Weiteren unterstützt es **kontinuierliche Sprache** und erlaubt dem Nutzer daher völlig frei und ohne Unterbrechungen zu sprechen.[110]

Die **Erkennungsgenauigkeit** dieser Software wird vom Hersteller mit 99% angegeben.[111] Auch im Praxistest steht Dragon dieser Angabe in nichts nach.[112] Die mobile Variante von Dragon unterstützt alle gängigen **Betriebssysteme** iOS, Android und Windows.[113]

Die mobile Variante von Dragon kann in eigene Applikationen **integriert** werden. Dies ist mit Hilfe einer API möglich, die vom Hersteller erworben werden kann. Ein

[106] Vgl. Nuance (o.J.e)
[107] Vgl. DAZ Diktiertechnik (o.J.)
[108] Vgl. Nuance (o.J.b)
[109] Vgl. Nuance (o.J.a)
[110] Vgl. Nuance (2014)
[111] Vgl. Nuance (o.J.d)
[112] Vgl. Spehr, M. (2014)
[113] Vgl. Nuance (o.J.g)

Web-Service ist nicht vorhanden.[114] Eine zusätzliche Funktion von Dragon ist die **Audio-Aufnahme** des gesprochenen Textes, um eventuell auftretende Fehler nachträglich ausbessern zu können.[115] Es können ebenfalls externe Mikrofone, insbesondere **Bluetooth-Mikrofone**, verwendet werden, um die Sprachqualität zu verbessern.[116] Jedoch ist die Benutzung eines **externen Mikrofons** keine Grundvoraussetzung, um die Software sinnvoll zu nutzen. Dragon erzielt auch mit dem internen Mikrofon des jeweiligen Gerätes eine ausreichende Erkennungsgenauigkeit.[117] Dragon erlaubt den Import und Export von eigens angelegtem **Vokabular**[118] und das Anlegen und Sichern von verschiedenen **Nutzerprofilen**, die ebenfalls auf neue Geräte importiert werden können.[119] Aufgrund der kommerziellen Vermarktung und der weiten Verbreitung von Dragon ist der Support für die Software sehr umfangreich. Für die diversen Softwarepakete sind ausreichende **Benutzerdokumentationen** vorhanden.[120] Bei gegebenenfalls auftretenden Fragen kann per E-Mail ein **Helpdesk** kontaktiert werden.[121] Es besteht darüber hinaus eine ausgeprägte Forenlandschaft mit einer aktiven **Support-Community**.[122] Dragon wird zurzeit **aktiv weiterentwickelt**. Momentan ist Version 13 auf dem Markt. In den letzten Jahren wurde etwa jährlich eine neue Version veröffentlicht.[123]

Xvoice / IBM ViaVoice

Xvoice ist eine Spracherkennungssoftware, welche die Diktierung von Sprache und die Programmsteuerung von verschiedenen Applikationen ermöglicht.[124] Sie ist unter der GNU General Public License verfügbar. Um die Sprache des Nutzers in Text umzuwandeln, wird die IBM ViaVoice Engine zur Spracherkennung benötigt, für die eine kostenpflichtige Lizenz erworben werden muss.[125] Xvoice macht sich sämtliche ViaVoice Bibliotheken zu Nutzen. Im Zuge dessen ist in den meisten Quellen über-

[114] Vgl. Nuance (o.J.c)
[115] Vgl. Nuance (2014)
[116] Vgl. Nuance (o.J.h)
[117] Vgl. Nuance (o.J.a)
[118] Vgl. Nuance (o.J.c)
[119] Vgl. Nuance (2014)
[120] Vgl. Nuance (o.J.g)
[121] Vgl. Nuance (o.J.f)
[122] Vgl. Informer Technologies Inc. (o.J.)
[123] Vgl. Wikipedia (2015)
[124] Vgl. Sourceforge (o.J.g)
[125] Vgl. ebenda

wiegend von ViaVoice als der eigentlichen Spracherkennungssoftware die Rede, weshalb diese Namensbezeichnung auch im Folgenden verwendet wird. ViaVoice bietet ein umfangreiches **Basisvokabular** von 64.000 Wörtern, welches sich um 260.000 Wörter individuell erweitern lässt.[126] Zum verwendeten **Sprachmodell** sind keinerlei Angaben zu finden.

Bei ViaVoice handelt es sich um eine **sprecheradaptive** Spracherkennungssoftware, d.h. die Software passt sich durch gezieltes Training der Aussprache des jeweiligen Sprechers an.

Hinsichtlich der **Sprachart** unterstützt ViaVoice kontinuierlich gesprochene Sprache, wodurch der Nutzer die Möglichkeit hat, das Diktieren ohne Pausen zwischen den einzelnen Wörtern durchzuführen.

Durch zusätzliches Training erreicht ViaVoice eine **Erkennungsgenauigkeit** von bis zu 98%.[127] Von Nutzern durchgeführte Tests ergeben, dass bereits das zehn- bis zwanzigminütige Vorlesen eines Textes äußerst effektiv sei, um die Erkennungsgenauigkeit erheblich zu steigern.[128]

Hinsichtlich des **Betriebssystems** sind IBM ViaVoice Desktop Produkte ursprünglich nur für Windows und Mac OS X bestimmt gewesen. Später wurde die IBM ViaVoice Engine jedoch auch für Linux verfügbar.

Hinsichtlich der **Integrationsfähigkeit** verzichtet ViaVoice sowohl auf jegliche APIs und Web-Services, als auch auf eine **Audio-Aufnahme** des diktierten Textes. Für die Diktierung selbst ist ein **externes Mikrofon** zwingend **erforderlich**. Aufgrund dessen wird ein Headset mit integriertem Mikrofon mitgeliefert, das über eine USB-Schnittstelle angeschlossen wird.[129] **Bluetooth** wird dagegen nicht unterstützt. Der **Import von Fachvokabular** ist prinzipiell möglich; angeboten wird eine Spezialversion von ViaVoice mit einem Fachvokabular für Ärzte und Juristen.[130] Erweitern lässt sich der Wortschatz von ViaVoice auch durch den Import von Dokumenten, indem diese nach unbekannten Wörtern durchsucht werden und für diese Begriffe die jeweilige Aussprache des Nutzers durch ein gezieltes Training erfasst wird.[131]

[126] Vgl. C. Crossman (2000)
[127] Vgl. G. Alwang (1999), S.174
[128] Vgl. ebenda, S.168
[129] Vgl. Pressrelations (1998)
[130] Vgl. Roth, W. (2009)
[131] Vgl. Alwang, G. (1999), S.174

Die **Sicherung von Nutzerprofilen** ist bei ViaVoice möglich, d.h. jeder Nutzer kann sein eigenes Profil, das auf seine individuelle Sprechweise angepasst ist, einrichten.[132]

Eine **Benutzerdokumentation** scheint in der Vergangenheit vorhanden gewesen zu sein. Jedoch ist der Link, der zu dieser führen soll, veraltet oder defekt und der Zugriff auf die Benutzerdokumentation nicht möglich.

Für ViaVoice respektive Xvoice existierte zur Jahrtausendwende eine breite **Support-Community**; jedoch ergeben Recherchen, dass diese seitdem sehr geschrumpft zu sein scheint, da themenbezogene Foreneinträge sehr veraltet sind.[133] Dies steht höchstwahrscheinlich in Zusammenhang mit der Tatsache, dass ViaVoice 2003 von IBM an Scansoft – heute Nuance – verkauft wurde.[134] Seither findet **keine aktive Weiterentwicklung** der Spracherkennungssoftware ViaVoice mehr statt. Der damalige Kooperationspartner Linguatec entwickelte 2009 in Zusammenarbeit mit Microsoft eine vollständig erneuerte Version von ViaVoice unter dem Namen *Linguatec Voice Pro,* wobei es sich hierbei um keine Open Source - Software handelt.[135]

Sphinx 4

Der **Wortschatz** bei Sphinx 4 umfasst im ursprünglichen CMU Pronouncing Dictionary 125.000 Wörter[136]. Jedoch lässt sich der Wortschatz nach Belieben erweitern[137] und hängt zudem vom genutzten Acoustic- und Language Model ab. Das **Sprachmodell** ist üblicherweise ein Trigramm[138]. Da dies jedoch ebenfalls auf dem genutzten Language Model basiert, ist hier die Nutzung von höheren N-Grammen bis zu Quantengrammen durchaus möglich- allerdings würde dies zu einem Mehraufwand führen.

[132] Vgl. Computer - merelewis.com (2010)
[133] Vgl. Sourceforge (o.J.g)
[134] Vgl. IBM (o.J.)
[135] Vgl. Roth, W. (2009)
[136] Vgl. Lenzo, K. (o.J.)
[137] Vgl. Kalra, P. (o.J.)
[138] Vgl. Sourceforge (o.J.a)

Sphinx 4 ist eine **sprecherunabhängige** Software[139], was die Genauigkeit der Spracherkennung insgesamt verschlechtert; daher erhält die Software in dieser Kategorie null Punkte.

Die **Sprachart** von Sphinx 4 ist kontinuierlich[140], d.h. die Spracheingabe im kontinuierlichen Fluss ohne Unterbrechungen zwischen einzelnen Wörtern wird unterstützt. Die **Erkennungsgenauigkeit** von Sphinx 4 wird von Seiten des Herstellers nicht benannt. Allerdings kann diese, basierend auf zwei Tests unabhängiger Quellen, mit 59 - 77% bewertet werden. Im ersten Test, der 2011 von Stephen Marquard mit Vorlesungsmaterialien der Yale Universität durchgeführt wurde, erreichte Sphinx 4 im Durchschnitt eine Word-Error-Rate (WER) von 41%. [141]

Der zweite Test wurde 2013 von Peter Grasch unter der Nutzung von VoxForge als Acoustic Model vorgenommen und ergab eine WER von 27,79%.[142] **Betriebssysteme**, die mit Sphinx 4 kompatibel sind, umfassen Solaris™, Mac OS X, Linux und Windows[143]. Da weder iOS noch Android unterstützt werden, erhält die Software in dieser Kategorie lediglich einen Punkt.

Bezüglich seiner **Integrationsfähigkeit** erhält Sphinx 4 hingegen die Höchstpunktzahl, da die Software sowohl APIs bereitstellt[144], als auch die Möglichkeit bietet, Web-Services einzubinden[145].

Sprachaufnahmen können gespeichert werden; daher erreicht Sphinx 4 in der Kategorie **Audio-Aufnahme** ebenfalls die Höchstpunktzahl. Für die Aktivierung der Audio-Aufnahme muss lediglich die in der Dokumentation bereits geschriebene Java-Klasse implementiert werden[146].

Auch **Bluetooth**-Mikrofone werden von Sphinx 4 unterstützt.[147] Generell ist die Nutzung eines **externen Mikrofons** nicht notwendig. Jedoch sind die hierzu erforderlichen Java- Klassen[148] verfügbar, da es sich bei Sphinx 4 um eine Desktop - Applika-

[139] Vgl. Hochstetter, J. (2008), S. 33
[140] Vgl. Marquard, S. (2011)
[141] Vgl. ebenda
[142] Vgl. Grasch, P. (2013)
[143] Vgl. Carnegie Mellon University (o.J.d)
[144] Vgl. Carnegie Mellon University (o.J.e)
[145] Vgl. Sourceforge (2009)
[146] Vgl. Sourceforge (o.J.c)
[147] Vgl. Voicerecog (2006)
[148] Vgl. Sourceforge (o.J.b)

tion handelt und somit herkömmlicherweise ein Headset o.Ä. verwendet wird. Genutzte **Wörterbücher** basieren auf dem gewählten Language Model und können, wie zuvor erwähnt, nach Belieben angepasst, verändert und importiert werden[149]. Da es sich bei Sphinx 4 um eine sprecherunabhängige Software handelt, ist die Sicherung von **Nutzerprofilen** nicht möglich.

Support zur Nutzung der Software gibt es in Form einer **Benutzerdokumentation**[150], einer aktiven Support-**Community**[151] und einer aktiven **Weiterentwicklung.**[152] Die letzte Aktualisierung ist im November 2014 erfolgt.

pocketSphinx

Der **Wortschatz** wird vom Nutzer eingepflegt und ist somit nicht zu bewerten. Zwar existieren vorgefertigte Sprachpakete, jedoch wird deren Umfang nicht genauer spezifiziert.
In Bezug auf das **Sprachmodell** gibt pocketSphinx dem Nutzer die Möglichkeit dieses bei der Konfiguration zu beeinflussen. Hier gibt es die Möglichkeit Quantengramme zu verwenden, weswegen für dieses Kriterium die höchste Punktzahl vergeben wird.[153]

PocketSphinx kann sprecheradaptiv eingesetzt werden und erreicht hinsichtlich der **Sprecherabhängigkeit** fünf Punkte.[154]

Für die **Sprachart** ist festzuhalten, dass der Umgang mit kontinuierlicher Sprache unterstützt wird.

Bezüglich der **Erkennungsgenauigkeit** macht der Herausgeber keine Angaben. Sie muss deshalb zur abschließenden Bewertung im Testprozess eigens ermittelt werden.
Im technischen Bereich, beginnend mit dem **Betriebssystem,** erzielt pocketSphinx die Bestnote, da alle gängigen mobilen Plattformen unterstützt werden.[155]

[149] Vgl. Carnegie Mellon University (o.J.a)
[150] Vgl. Carnegie Mellon University (o.J.f)
[151] Vgl. Sourceforge (o.J.e)
[152] Vgl. Sourceforge (o.J.d)
[153] Vgl. Sourceforge (2014a)
[154] Vgl. Carnegie Mellon University (o.J.g)
[155] Vgl. Carnegie Mellon University (o.J.b)

Zusätzlich wird sowohl durch die Bereitstellung von APIs[156] als auch durch Möglichkeiten der Implementierung als Webservice höchste **Integrationsfähigkeit** gewährleistet.[157]

Die **Audio-Aufnahme** neben der Umsetzung als Text ist nicht möglich. **Bluetooth** wird unterstützt, genauso wie interne **Mikrofone** der Endgeräte, da pocketSphinx sich des zu Grunde liegenden Audio-Interfaces des jeweiligen Gerätes bedient.

Wörterbücher können, wie eingangs erwähnt, vom Nutzer direkt bei der Konfiguration **importiert** werden und stehen als Dateien zur Verfügung, die einen einfachen Export ermöglichen.[158]

Nutzerprofile können nicht angelegt werden.

Im Support Bereich gibt es zum einen eine **Online-Benutzerdokumentation,** zum anderen ein **Support Forum.** Die letzte Version ist vor etwa zwölf Monaten erschienen. Dies zeugt demnach von einer **aktiven Weiterentwicklung** der Software**.**[159]

3.3.1. Vergleichende Analyse

Der **Wortschatz** ist nur bei zweien der insgesamt vier Programme, Dragon und Xvoice, eindeutig auszumachen. Hier liegen beide mit einer Wörterzahl von über 250.000 Wörtern (Dragon) und 260.000 Wörtern (Xvoice) ungefähr gleich auf. Die beiden weiteren Programme Sphinx 4 und pocketSphinx haben jedoch beide ein Vokabular, das durch den Nutzer beliebig erweitert werden kann: Sphinx 4 weist ein ursprüngliches Vokabular von 125.000 Wörtern auf, pocketSphinx dagegen kann mit vorgefertigten Sprachpaketen aufwarten, die allerdings völlig unterschiedliche Fachvokabulare aufweisen.

Das verwendete **Sprachmodell** kann bei Xvoice nicht ausgemacht werden. Dragon und pocketSphinx erreichen hier mit verfügbaren Quantengrammen jeweils die Höchstpunktzahl von 5 Punkten, während Sphinx 4 in dieser Kategorie 3 Punkte er-

[156] Vgl. Sourceforge (2014b)
[157] Vgl. Github (2014)
[158] Vgl. Carnegie Mellon University (o.J.c)
[159] Vgl. Carnegie Mellon University (o.J.h)

langt. Durch die hohe Gewichtung dieses Kriteriums werden die Punktzahlen noch einmal mit 5 multipliziert.

Xvoice, Dragon und pocketSphinx sind **sprecheradaptive** Systeme, wodurch sie in Bezug auf die **Sprecherabhängigkeit** die Höchstpunktzahl erhalten. Sphinx 4 hingegen ist ein sprecherunabhängiges System und erreicht folglich keinen Punkt in dieser von der .Versicherung als Auftraggeber als sehr wichtig eingestuften Kategorie.
Kontinuierliche Sprache wird von allen anhand der Kriterien des Kriterienkataloges untersuchten Spracherkennungsprogrammen unterstützt.

Bei der **Erkennungsgenauigkeit** schneidet Dragon mit 99% am besten ab; hierauf folgt Xvoice mit einer Genauigkeit von bis zu 98%. Sphinx 4 bleibt selbst bei eigenen Höchstwerten mit einer Erkennungsgenauigkeit von unter 80% weit zurück; zu pocketSphinx sind keine Informationen hinsichtlich dieses Kriteriums verfügbar. Sowohl bei Dragon als auch bei pocketSphinx werden in Bezug auf das **Betriebssystem** die Plattformen iOS, Android und Windows unterstützt. Sphinx 4 sowie Xvoice unterstützen lediglich Windows und weitere Betriebssysteme, die nicht als vom Auftraggeber gewünscht herausgestellt wurden.

In Bezug auf die **Integrationsfähigkeit** bilden pocketSphinx und Sphinx 4 die am besten auf die Präferenzen der .Versicherung zugeschnittenen Systeme, da sie sowohl APIs bereitstellen, als auch als Webservices nutzbar sind. Das kommerzielle Vergleichsprodukt Dragon ist nur als Webservice verfügbar. Zur Integrationsfähigkeit von Xvoice konnten keine Angaben ermittelt werden.

Audio-Aufnahmen sind bei den Spracherkennungsprogrammen Dragon und Sphinx 4 möglich. Bei den beiden Vergleichsprodukten werden keine Audio- Aufnahmen erstellt.
Bluetooth wird bei den Systemen pocketSphinx, Sphinx 4 sowie Dragon unterstützt; bei Xvoice ist dies nicht der Fall.

Ein **externes Mikrofon** ist lediglich bei Xvoice erforderlich.
Der **Import und Export** von (Fach-)Vokabular ist bei allen analysierten Systemen möglich.
Nutzerprofile können nur bei den Programmen Xvoice und Dragon angelegt werden.

Bei Xvoice ist im Vergleich zu den anderen drei Systemen keine **Benutzerdokumentation** einsehbar.

Alle Systeme haben eine **Support-Community**. Diese scheint jedoch bei Xvoice seit der Jahrtausendwende entweder stark geschrumpft zu sein oder sich komplett aufgelöst zu haben. Insgesamt besticht bei diesem Kriterium das kommerzielle System Dragon durch ein zusätzliches **Helpdesk**, das eigens für Nutzer dieses Produktes besteht und an das sich diese per E-Mail direkt wenden können.
Nicht nur in Bezug auf die Support-Community hat sich bei Xvoice seit der Jahrtausendwende wenig getan; auch eine **aktive Weiterentwicklung** dieses Systems findet seit 2003 nicht mehr statt, da die Spracherkennungssoftware in diesem Jahr an das Unternehmen ScanSoft (heute Nuance Communications, welches auch der Entwickler des Produktes Dragon ist) verkauft wurde. Dragon wird aktiv weiterentwickelt; im Schnitt wird einmal pro Jahr eine neue Version dieses Systems veröffentlicht. Die letzte Version von pocketSphinx erschien vor circa zwölf Monaten, während Sphinx 4 zuletzt im November 2014 aktualisiert wurde.
In der **Gesamtwertung** (siehe Kriterienkatalog) belegt die kommerzielle Software Dragon mit einer Gesamtpunktzahl von 246 den ersten Platz. Da der Auftraggeber jedoch eine Open Source - Lösung sucht (siehe 2.7.1. Anforderungsprofil unter dem Oberbegriff *Softwarelizenz*), stellt Dragon kein für die .Versicherung geeignetes Produkt dar. Laut Kriterienkatalog belegt die Spracherkennungssoftware pocketSphinx mit einer Gesamtpunktzahl von 169 Punkten den zweiten Platz, Sphinx 4 mit 159 Punkten den dritten und Xvoice mit 128 Punkten den vierten Platz.

3.4. Testprozess

Der Testprozess stellt den letzten Schritt in der Analyse der Spracherkennungssoftware dar. Zum Testen der Software wird ein Prozess definiert, der einer bestimmten Methodik folgt und für jede Anwendung gleich verläuft, um Similaritäten und Unterschiede festzustellen.

Da Dragon NaturallySpeaking ausschließlich kommerziell zu erwerben ist, wird für den Test - in Absprache mit der .Versicherung - die Software **Dragon Dictation** benutzt werden. Diese greift auf dasselbe Back-End zu, ist aber als Applikation kostenlos zum Download verfügbar und wird daher für den Test eingesetzt.

Hinsichtlich verfügbarer Open Source - Produkte wird **pocketSphinx** als zweitplatzierte Software nach der Auswertung anhand des Kriterienkataloges zum Test herangezogen.

Zwar sollte auch Sphinx 4 als weitere Open Source - Lösung betrachtet werden, jedoch stammen Sphinx 4 und pocketSphinx von demselben Hersteller und unterscheiden sich lediglich in den Plattformen, auf denen sie zur Anwendung gedacht sind. Da pocketSphinx der Anwendung auf mobilen Endgeräten dient und die .Versicherung die Open Source - Software später auf Tablets nutzen möchte, wird diese als Vertreter der Sphinx 4 - Software im Test betrachtet.

Durch die Gleichsetzung von Sphinx 4 und pocketSphinx könnte zwar noch ViaVoice als weitere Open Source - Software getestet werden; jedoch hat sich dieses Spracherkennungsprogramm als stark veraltet erwiesen und kommt aufgrund mangelnder aktiver Weiterentwicklung der Software für die aktive Nutzung im Vertrieb der .Versicherung nicht infrage.

3.4.1 Methodik

Der Test der verschiedenen Spracherkennungsprogramme erfolgt durch eine (Demo-)Installation der Anwendungen und deren aktiven Gebrauch in einer kontrollierten Testumgebung. Ergänzend hierzu werden die Testergebnisse in einem Testkatalog (siehe Anhang 3) festgehalten.

Ein beispielhafter Schadensbericht soll von fünf Testpersonen gelesen und zur Spracherkennung eingegeben werden. Dies soll sowohl im Innenbereich ohne Lärmbelastung, als auch mit inszeniert höherer Lärmbelastung (bspw. Musik im Hintergrund) erfolgen.

Die Ausführung des Testes erfolgt auf einem iPhone 4 (Betriebssystem iOS) für Dragon Dictation und auf einem Nexus 4 (Betriebssystem Android) für pocketSphinx. Die Spracheingaben werden an demselben Tag sowie am selben Ort durchgeführt, um hinsichtlich der Ergebnisse der Testpersonen eine optimale Vergleichbarkeit für die Spracherkennungsprogramme herzustellen.

Die anschließende Bewertung erfolgt primär anhand der Fehleranzahl der genutzten Sprachsoftware, ebenso werden die Dauer der Verarbeitung, die CPU-Last und RAM-Nutzung ausgewertet.

Der Text, der von den Testpersonen sowohl bei pocketSphinx als auch bei Dragon Dictation zum Test diktiert wird, beinhaltet drei Sätze des folgenden beispielhaften Schadensberichtes:

“Der Wagen hat eine Delle hinten rechts am Tankdeckel.

Es sind erhebliche Wasserschäden in der Küche, im Wohnzimmer und im Bad des Hauses.

Der Kostenvoranschlag beläuft sich auf 543 Euro und 49 Cent.”

3.4.2 Testergebnisse

Als Tag der Testdurchführung wird der 17. Januar 2015 festgelegt. Die zu diktierenden Sätze dürfen durch die Testpersonen bereits vor der Spracheingabe durchgelesen werden. Interpunktionen werden beim Diktat ausdrücklich genannt. Sowohl Dragon Dictation als auch pocketSphinx werden zunächst ohne Lärmbelastung in einem geschlossenen Raum getestet.

Danach wird der Prozess mit demselben Text wiederholt, jedoch mit erhöhter Lärmbelastung. Diese wird durch Musik und Unterhaltungen im Hintergrund und sonstigen Nebengeräuschen simuliert. In beiden Durchläufen werden die Fehleranzahl der Software, die Dauer der Verarbeitung in Sekunden, die CPU-Last und die RAM-Nutzung notiert.

Die **CPU-Last** beträgt für pocketSphinx im Test auf einem Nexus 4 40%. Die CPU-Last bei Dragon Dictation auf einem iPhone 4 beträgt hingegen 100%. Hierbei muss jedoch beachtet werden, dass sich die verfügbare Rechenleistung des iPhone 4 auf einen 1 GHz Prozessor mit 1-Core beschränkt, während das Nexus 4 mit einem 1,5 GHz Prozessor und Quad-Core Technologie aufwartet. Es ist nicht möglich, beide Softwarelösungen auf demselben Gerät zu testen, da Dragon Dictation nur für iOS und pocketSphinx nur für Android erhältlich ist. Da die CPU-Last zu

sehr von den Spezifikationen der benutzten Geräte abhängt und sich diese Problematik nicht beheben lässt, wird sie im Gesamtvergleich der Software nicht weiter betrachtet.

Die **RAM-Nutzung** kann im Test nicht verglichen werden, da diese lediglich für pocketSphinx mit 100 MB festgestellt werden kann; das iPhone 4 ermöglicht keine Einsicht in die Auslastung des Arbeitsspeichers.

Abb. 2: Vergleich der Verarbeitungsdauer

Die unter der obigen Abbildung 2 für die beiden getesteten Programme anhand der fünf Testpersonen in Sekunden dargestellte **Dauer der Verarbeitung** ist bei beiden Anwendungen sehr kurz. Während Dragon Dictation das Ergebnis der Spracheingabe nach zwei Sekunden anzeigt, beträgt die Verarbeitungsdauer bei pocketSphinx lediglich eine Sekunde.

In dieser Kategorie schneidet pocketSphinx demnach besser ab, wobei der Unterschied von einer Sekunde auch in der praktischen Anwendung im Vertrieb der .Versicherung keine nennenswerten Schwierigkeiten verursachen würde.

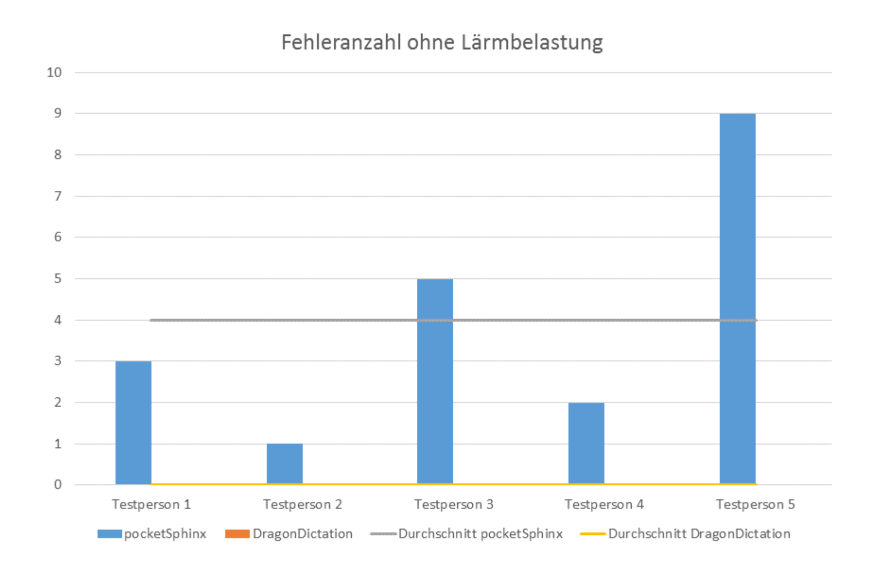


Abb. 3: Fehleranzahl ohne Lärmbelastung

Beim Test von Dragon Dictation ohne Lärmbelastung, der in der obigen Abbildung 3 dargestellt ist, treten bei keinem der Tester **Fehler in der Spracherkennung** auf; jedes Wort wird von der Open Source - Software korrekt erkannt. Die durchschnittliche Fehleranzahl ist demnach null.

Im Test mit Lärmbelastung sehen die Ergebnisse anders aus: Hier liegen bei drei von fünf Testpersonen Fehler im Text vor. Mit einer Anzahl von drei, fünf und acht Fehlern weist Dragon eine durchschnittliche Fehleranzahl von 3,2 Fehlern auf. Der Test von pocketSphinx ohne Lärmbelastung verläuft bei keinem der Tester fehlerfrei. Die Fehleranzahl variiert von einem bis hin zu neun Fehlern; die durchschnittliche Fehleranzahl beträgt vier Fehler.

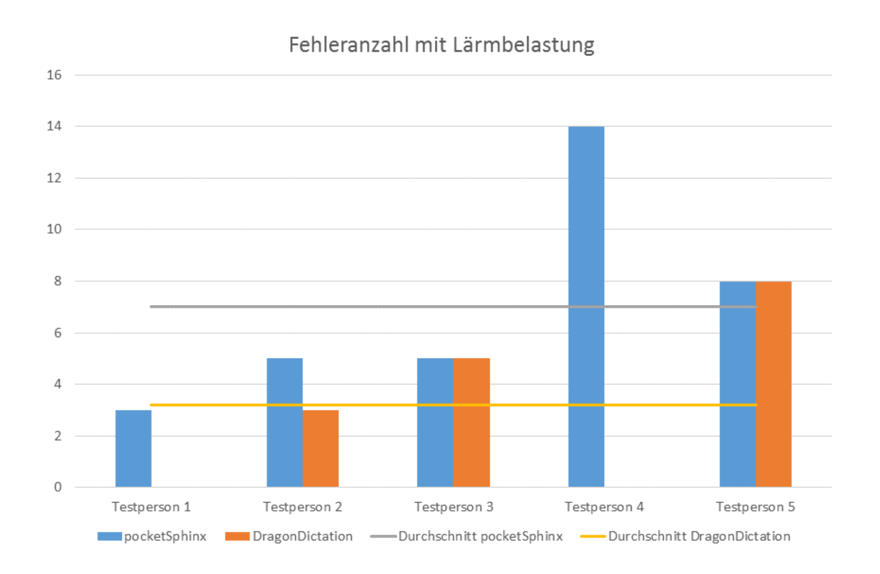


Abb. 4: Fehleranzahl mit Lärmbelastung

Mit einer erhöhten Lärmbelastung steigt, wie in der Abbildung 4 ersichtlich, jedoch auch bei der Software pocketSphinx die Zahl der falsch erkannten Wörter. Die Anzahl beträgt drei bis vierzehn Fehler, der Durchschnitt liegt somit bei sieben Fehlern. Insgesamt weist Dragon Dictation (mit und ohne Lärmbelastung) eine durchschnittliche Fehlermenge von 1,6 auf, während pocketSphinx im Schnitt 5,5 Falscherkennungen pro Texteingabe zu verzeichnen hat.

Bei pocketSphinx ist zu beobachten, dass einzelne Wörter bei der Verschriftlichung zum Teil zusammengezogen werden und dadurch der Sinn verfälscht wird. Beispielsweise wird der Satzteil “im Bad des Hauses” zweimal als “Badehaus” interpretiert, einmalig je als “Rathaus” und “Wartehalle”. Die Tatsache, dass das Wort “Bad” mit einem “d” endet, während das darauffolgende Wort “des” mit diesem Buchstaben beginnt, führt vermutlich dazu, dass sie bei der Diktierung zusammengezogen und als eine Einheit verstanden werden. Demnach kann es bei erhöhter Lärmbelastung hilfreich sein, die Wörter bewusst einzeln auszusprechen.
Der zweithäufigste Fehler ist das vollständige Fehlen ganzer Wörter und Satzzeichen; dies betrifft vor allem die durchgeführten Tests mit Hintergrundgeräuschen. Beispielhaft wird im Folgenden der interpretierte Text von pocketSphinx unter Lärmbelastung, diktiert durch Testperson 4, wiedergegeben:
“Der Wagen hat eine Delle im Tank Deckel.

Wasserschäden in der Küche, im Hause

Der Kostenvoranschlag beläuft sich auf 573 Euro und 49 Cent.”

Wie erwähnt ist hier das gänzliche Fehlen einiger Teile des Originaltextes problematisch, da vor allem der zweiten Satz stark verfremdet wird. Nichtsdestotrotz ist die Leistung von pocketSphinx (ohne Lärmbelastung) insgesamt sehr gut: die Dauer der Verarbeitung ist schneller als bei Dragon Dictation und auch die Fehleranzahl ist gering im Vergleich zu der Gesamtzahl der Wörter: die Word-Error-Rate beträgt lediglich 10,8%.

4 Schlussbetrachtung und Empfehlung

Momentan werden auf dem Markt zahlreiche Open Source - Spracherkennungsprogramme angeboten; die meisten davon befinden sich jedoch noch im Teststatus und sind in ihrer derzeitigen Form nicht für eine kommerzielle Nutzung geeignet. Aus dieser Masse an Open Source - Programmen verbleiben nach Filterung durch das Anforderungsprofil drei Anwendungen, die zusammen mit Dragon Naturally-Speaking durch den Kriterienkatalog weiter analysiert werden. Durch Auswertung der Analyse anhand des Kriterienkataloges stellen sich unter den Open Source - Produkten pocketSphinx und Sphinx 4 eindeutig heraus, während ViaVoice als verbleibendes Anwendungsprogramm nicht zur aktiven Verwendung in Betracht kommt, da es seit einigen Jahren weder aktualisiert wird, noch eine nötige Support-Community aufweist.

Somit werden letztlich die Open Source - Software pocketSphinx und das kommerzielle Produkt Dragon Dictation im Test verglichen. Obwohl Dragon bezüglich der Word-Error-Rate die besseren Ergebnisse erzielt und vor allem bei Störgeräuschen ganz klar der Software pocketSphinx überlegen ist, schneidet pocketSphinx nicht nennenswert schlechter ab: Eine WER von 10,8% ist in Anbetracht der Tatsache, dass die Tester nicht auf den Gebrauch einer Spracherkennungssoftware geschult waren und dass es sich hierbei um erste Versuche der Nutzung handelte, vergleichsweise gut.

Obgleich Dragon als Marktführer der kommerziellen Produkte auch in diesem Test ganz klar als Sieger hervorgeht, liefert pocketSphinx bei ruhiger Umgebung nahezu

genauso gute Ergebnisse, weshalb diese Software auch eindeutig als Open Source - Produkt zur Spracherkennung empfohlen wird.

Als Einschränkung wird allerdings zugleich empfohlen, die Software vor allem in einer ruhigeren Umgebung zu nutzen, da die Filterung von Störgeräuschen im Hintergrund derzeit als problematisch zu bewerten ist.

Nichtsdestotrotz liefert pocketSphinx eine sehr gute Open Source - Alternative zu dem kommerziellen Produkt. Es überzeugt sowohl durch das vergleichsweise gute Abschneiden bei den im Kriterienkatalog bewerteten Spezifikationen mit einer Punktzahl von 169 Punkten, als auch im direkten Test.

Es bleibt abzuwarten, was in Zukunft an Spracherkennungssoftware entwickelt werden wird und auch wie sich der Open Source - Markt für Software generell weiterentwickelt. Solange es eine Zeit- und damit Kostenersparnis darstellt, mit Spracheingaben statt mit manuellen schriftlichen Eingaben zum gewünschten Ergebnis zu kommen, ist in diesem Bereich mit Sicherheit auch in den nächsten Jahren mit neuen Erkenntnissen und Fortschritten zu rechnen.

Vielversprechend ist auch die Forschung der Defense Advanced Research Projects Agency (DARPA), die zurzeit mit drei Forscherteams an einem Programm zum Einlesen fremdsprachiger Nachrichten und einer direkten Übersetzung in die gewünschte Sprache arbeitet (vgl. 2.2.). Da diese Technologie zukünftig auch im Militär eingesetzt werden soll, sind überaus spannende Entwicklungen im Bereich der Spracherkennung im Allgemeinen zu erwarten, die sich von der reinen Umwandlung von Sprache in Text künftig immer weiter in Richtung eines Sprachverstehens entwickeln wird. In der Zukunft soll demnach nicht nur das Gesprochene an sich, sondern zusätzlich die Bedeutung des Gesagten von der Software verstanden werden können.[160] Einige Forscher gehen sogar so weit zu behaupten, dass die heutige Spracherkennung eine geradewegs direkte Verbindung von moderner Computertechnologie zu künstlicher Intelligenz sei.[161]

[160] Vgl. Grabianowski, E. (o.J.e)
[161] Vgl. ebenda

Anhang

Anhangverzeichnis

Software	Ausschluss (ja/nein)	Grund des Ausschlusses anhand des Anforderungsprofils (zu erfüllende Mindestvoraussetzung des Auftraggebers)			
		Softwarelizenz (Open Source)	Ort der Verarbeitung (Front-End Lösung/ Back-End-Lösung mit eigenem Server)	Spracherkennungstyp (Diktiersystem-Funktion)	Sprache (Deutsch)
CVoiceControl/ kVoiceControl	ja				nur Englisch
OpenEars	ja				nur Englisch, Spanisch
Emacspeak	ja				nur Englisch
eSpeak	ja			nur Text to Speech möglich	
Festival Speech Synthesis System	ja				nicht auf Deutsch verfügbar
Gvoice	ja	Kommerzielle Software			

Anhang 1-1: Anforderungsprofil Analyse

HDecode (Hidden Markov Model Toolkit)	ja			nur ein Toolkit; keine Diktiersystem- Funktion	
ISIP	ja				nur auf Englisch
Ivona	ja			nur Text to Speech	
Jialong He's Speech Recognition Research Tool	ja			ein Forschungstool; keine Spracherkennung möglich	
Julius	ja				nur Englisch und Japanisch
Kaldi	ja				nur Englisch
Modular Audio Recognition Framework	ja			keine Spracherkennung, vielmehr eine in Java geschriebene Open Source-Forschungsplattform	
Myers' Hidden Markov Model Software	ja			Keine Spracherkennungs-software ; Code dient nur der Erstellung eines experimentellen Spracherkennungs-systems	

Anhang 1-2: Anforderungsprofil Analyse

NICO ANN Toolkit	ja			nur Text to Speech möglich	
Open Mind Speech	ja			es ist zurzeit noch nicht vollständig beriebsbereit und daher nur zur Weiterentwicklung geeignet	
pocketSphinx	nein				
RWTH ASR	ja	Die Nutzung zu kommerziellen Zwecken ist nicht erlaubt			
Simon	ja			nur zur Sprachsteuerung von Tastatur und Maus einsetzbar	
Sphinx 4 + Vox Forge	nein				
sphinxkeys	ja			nur zur Sprachsteuerung von Tastatur und Maus einsetzbar	
XVoice	nein				

Anhang 1-3: Anforderungsprofil Analyse

Kriterium	Bewertungsmaßstab	Gewichtungsfaktor	Software			
			pocketSphinx Kriterienpunktzahl*	Dragon NaturallySpeaking 13 Kriterienpunktzahl*	Sphinx 4 Kriterienpunktzahl*	Xvoice / ViaVoice Kriterienpunktzahl*
Wortschatz	> 220.000 Wörter: 5 Punkte 220.000 bis > 180.000 Wörter: 4 Punkte 180.000 bis > 140.000 Wörter: 3 Punkte 140.000 bis > 100.000 Wörter: 2 Punkte	3	0	15	6	15
Sprachmodell	Quantengramm: 5 Punkte Tetragramm: 4 Punkte Trigramm: 3 Punkte Bigramm: 2 Punkte	5	25	25	15	N/A
Sprecherabhängigkeit	Sprecheradaptiv: 5 Punkte Sprecherabhängig: 3 Punkte Sprecherunabhängig: 0 Punkte	5	25	25	0	25
Sprachart	Kontinuierlich: 5 Punkte Andere: 0 Punkte	3	15	15	15	15
Erkennungsgenauigkeit	über 97%: 5 Punkte 97% bis > 95%: 4 Punkte 95% bis > 93%: 3 Punkte 93% bis > 90%: 2 Punkte 90% bis 80%: 1 Punkt < 80%: 0 Punkte	5	0	25	0	20

Anhang 2-1: Kriterienkatalog

Betriebssystem	IOS, Android & Windows: 5 Punkte mind. IOS: 3 Punkte mind. Android oder Windows: 1 Punkt	1	5	5	1	1
Integrationsfähigkeit	API & Web- Service: 5 Punkte nur Web- Service: 3 Punkte nur API: 2 Punkte keines von beidem: 0 Punkte	3	15	9	15	0
Audio-Aufnahme	ja: 5 Punkte nein: 0 Punkte	3	0	15	15	0
Bluetooth-Unterstützung	ja: 5 Punkte nein: 0 Punkte	2	10	10	10	0
Externes Mikrofon erforderlich	nein: 5 Punkte ja: 0 Punkte	2	10	10	10	0
Wörterbuch Import/Export (Fachvokabular)	ja: 5 Punkte nein: 0 Punkte	4	20	20	20	20
Sicherung von Nutzerprofilen	ja: 5 Punkte nein: 0 Punkte	4	0	20	0	20

Anhang 2-2: Kriterienkatalog

Benutzerdokumentation vorhanden	ja: 5 Punkte nein: 0 Punkte	4	20	20	20	0
Support- Community	Helpdesk/ Telefonischer Support: 5 Punkte Support- Community: 3 Punkte Keines von beidem: 0 Punkte	4	12	20	12	12
Aktive Weiterentwicklung	Release- Zyklus ≤ 3 Monate: 5 Punkte Release- Zyklus 3 Monate bis 12 Monate: 3 Punkte Release- Zyklus > 12 Monate: 0 Punkte	4	12	12	20	0
Gesamtpunktzahl für die Software			169	246	159	128
Positionierung der Software			2.	1.	3.	4.

*Kriterienpunktzahl = Bewertungsmaßstab * Gewichtungsfaktor

Anhang 2-3: Kriterienkatalog

Testkategorie	Software	Umgebung	Testperson	Ergebnis
Fehleranzahl	pocketSphinx	Ohne Lärmbelastung	1	3
			2	1
			3	5
			4	2
			5	9
		Mit Lärmbelastung	1	3
			2	5
			3	5
			4	14
			5	8

Anhang 3-1: Testkatalog (Fehleranzahl pocketSphinx)

Fehleranzahl	Dragon Dictation	Ohne Lärmbelastung	1	0
			2	0
			3	0
			4	0
			5	0
		Mit Lärmbelastung	1	0
			2	3
			3	5
			4	0
			5	8

Anhang 3-2: Testkatalog (Fehleranzahl Dragon Dictation)

Dauer der Verarbeitung (in Sek.)	pocketSphinx	Ohne Lärmbelastung	1	1
			2	1
			3	1
			4	1
			5	1
		Mit Lärmbelastung	1	1
			2	1
			3	1
			4	1
			5	1
	Dragon Dictation	Ohne Lärmbelastung	1	2
			2	2
			3	2
			4	2
			5	2
		Mit Lärmbelastung	1	2
			2	2
			3	2
			4	2
			5	2

Anhang 3-3: Testkatalog (Dauer der Verarbeitung)

CPU Last	pocketSphinx	Ohne Lärmbelastung	1	40%
			2	40%
			3	40%
			4	40%
			5	40%
		Mit Lärmbelastung	1	40%
			2	40%
			3	40%
			4	40%
			5	40%
	Dragon Dictation	Ohne Lärmbelastung	1	100%
			2	100%
			3	100%
			4	100%
			5	100%
		Mit Lärmbelastung	1	100%
			2	100%
			3	100%
			4	100%
			5	100%

Anhang 3-4: Testkatalog (CPU Last)

RAM Nutzung	pocketSphinx	Ohne Lärmbelastung	1	100 MB
			2	100 MB
			3	100 MB
			4	100 MB
			5	100 MB
		Mit Lärmbelastung	1	100 MB
			2	100 MB
			3	100 MB
			4	100 MB
			5	100 MB
	Dragon Dictation	Ohne Lärmbelastung	1	nicht feststellbar
			2	nicht feststellbar
			3	nicht feststellbar
			4	nicht feststellbar
			5	nicht feststellbar
		Mit Lärmbelastung	1	nicht feststellbar
			2	nicht feststellbar
			3	nicht feststellbar
			4	nicht feststellbar
			5	nicht feststellbar

Anhang 3-5: Testkatalog (RAM Nutzung)

Liste Spracherkennungssoftware

Open Source

- CVoiceControl/kVoiceControl (only execution of Linux commands)
- Ears
- Emacspeak
- eSpeak
- Festival Speech Synthesis System
- GVoice
- HDecode
- ISIP
- Ivona
- Jialong He's Speech Recognition Research Tool
- Julius
- Kaldi
- Modular Audio Recognition Framework
- Myers' Hidden Markov Model Software
- NICO ANN Toolkit
- Open Mind Speech (not completely operational)
- pocketSphinx
- RWTH ASR
- Simon
- Sphinx-4
- sphinxkeys
- XVoice

Anhang 4: Liste der OS Spracherkennungssoftware

Quellenverzeichnisse

Literaturverzeichnis

- Bruns, K.; Ackermann, R. (2005): Taschenbuch der Medieninformatik. Mit 39 Tabellen. München [u.a.]: Fachbuchverl. Leipzig im Carl-Hanser-Verl.
- Dworatschek, S. (1989): Grundlagen der Datenverarbeitung. 8., durchgesehene Aufl. Berlin, New York: de Gruyter (De-Gruyter-Lehrbuch).
- Euler, S. (2006): Grundkurs Spracherkennung. Vom Sprachsignal zum Dialog; Grundlagen und Anwendung verstehen; mit praktischen Übungen. 1. Aufl. Wiesbaden: Vieweg (Computational intelligence).
- Greg A. (1999): Speech Recognition, in: PC Magazin, Dezember, pp. 167–182, https://books.google.co.uk/books?id=90OvoBUqQoIC&pg=PA174&lpg=PA174&dq=PC+Magazine%2C+viavoice&source=bl&ots=QrmA9L4wlp&sig=AtdcUryKln16tLI_e71IuNwPM9Y&hl=de&sa=X&ei=hXS6VMfQOouxUY3-gbgO&ved=0CCEQ6AEwAA#v=onepage&q=PC%20Magazine%2C%20viavoice&f=false, Abruf 22.12.2014.
- Henning, P. A. (2007): Taschenbuch Multimedia. Mit zahlreichen Tabellen. [Elektronische Ressource], 4, aktualisierte Aufl. München: Fachbuchverl. Leipzig im Carl-Hanser-Verl.
- Zumalt, R. J. (2005): Voice Recognition Technology: Has It Come Of Age? in Information Technology and Libraries 24 (4), pp. 180–185. https://136.167.2.46/ojs/index.php/ital/article/view/3382/2992, Abruf: 19.12.2014.
- Hochstetter, J (2008): Spracheingabe und Sprachausgabe für mobile Serviceroboter. Hochschule Ravensburg-Weingarten.
- Fellbaum, K. (2012): Sprachverarbeitung und Sprachübertragung. 2. Aufl. Berlin: Springer.

- Lee, K. (1989): Automatic Speech Recognition. The Development of the SPHINX System. Boston, MA: Springer US.
- Lehner, F. (1994): Software-Dokumentation und Messung der Dokumentationsqualität. München, Wien: Hanser.
- Pfister, B./ Kaufmann, T. (2008): Sprachverarbeitung. Grundlagen und Methoden der Sprachsynthese und Spracherkennung. Berlin, Heidelberg: Springer.

Internetverzeichnis

- Appelt, J. (o.J.): Alles über Arbeitsspeicher. Computer Bild. http://www.computerbild.de/artikel/cb-Ratgeber-PC-Hardware-Alles-ueber-Arbeitsspeicher-17097.html, Abruf: 13.12.201.
- BBC (2009): Voice-Recognition Software – An Introduction. http://www.bbc.co.uk/accessibility/guides/factsheets/factsheet_VR_intro.pdf, Abruf: 20.12.2014.
- Bednar, J. (2014): Word Up. As Speech-recognition Technology Improves, More Applications Emerge Februar. http://businesswest.com/blog/word-up/, Abruf: 20.12.2014.
- Business Dictionary (o.J.): speech recognition. http://www.businessdictionary.com/definition/speech-recognition.html, Abruf: 21.12.2014.
- Carnegie Mellon University (o.J.a): Building Language Model. http://cmusphinx.sourceforge.net/wiki/tutoriallm, Abruf: 20.12.2014.
- Carnegie Mellon University (o.J.b): Building on Mobile. http://cmusphinx.sourceforge.net/wiki/building, Abruf: 20.12.2014.
- Carnegie Mellon University (o.J.c): Generating a dictionary. http://cmusphinx.sourceforge.net/wiki/tutorialdict, Abruf: 23.12.2014.
- Carnegie Mellon University (o.J.d): How to Build and Run Sphinx4. http://cmusphinx.sourceforge.net/wiki/sphinx4:howtobuildand_run_sphinx4, Abruf: 22.12.2014.

- Carnegie Mellon University (o.J.e): Sphinx4. http://cmusphinx.sourceforge.net/wiki/sphinx4:webhome, Abruf: 12.12.2014.
- Carnegie Mellon University (o.J.f): Start User Documentation. http://cmusphinx.sourceforge.net/wiki/, Abruf: 20.12.2014.
- Carnegie Mellon University (o.J.g): Training Acoustic Model For CMUSphinx. http://cmusphinx.sourceforge.net/wiki/tutorialam, Abruf: 20.12.2014.
- Carnegie Mellon University (o.J.h): US English acoustic model update. http://cmusphinx.sourceforge.net/2015/01/us-english-acoustic-model-update/, Abruf: 02.01.2015.
- Computer - merelewis.com (2010): So verwenden Sie IBM ViaVoice Software. http://www.merelewis.com/so-verwenden-sie-ibm-viavoice-software/, Abruf: 02.01.2015.
- Computerwoche (2012): Von IBM Shoebox bis Siri. http://www.computerwoche.de/a/von-ibm-shoebox-bis-siri,2510153, Abruf: 23.01.2015.
- Crossman, C. (2000): Ibm's Software Lets Mac Users Speak Up. Chicago Tribune. http://articles.chicagotribune.com/2000-01-03/business/0001030177_1_mac-os-imac-microsoft-word, Abruf: 21.12.0214.
- DAZ Diktiertechnik (o.J.): Dragon NaturallySpeaking® 13 Professional Solutions. https://diktiertechnik.de/Spracherkennung2.php, Abruf: 10.12.2014.
- dc-Systeme Informatik GmbH (2012): Spracherkennung. http://www.dc-systeme.de/index.php?s=spracherkennung, Abruf: 19.12.2014.
- Duden (o.J.): Sprache. http://www.duden.de/rechtschreibung/Sprache#Bedeutung1, Abruf: 21.12.2014.
- FAZ (2013): Mikrofone für Spracherkennung. Kabel, Bluetooth oder Dect. http://www.faz.net/aktuell/technik-motor/computer-internet/mikrofone-fuer-spracherkennung-kabel-bluetooth-oder-dect-12140541/beste-akustik-aber-mit-kabel-12145506.html, Abruf: 14.12.2014.

- FH Wedel (o.J.): Technische Problemstellung. http://www.fh-wedel.de/~si/seminare/ss01/Ausarbeitung/a.sprache/gdlgsprerk21.htm, Abruf: 27.11.2014.
- Germesin, S. (2006): Ausarbeitung des Vortrags Ausarbeitung des Vortrags über das Thema Spracherkennung. Deutschen Forschungsinstitut für Künstliche Intelligenz. Saarbrücken. http://www.dfki.de/~kipp/seminar/writeups/Sebastian_ivc.pdf, Abruf: 22.12.2014.
- Github (2014): Ruby-based web service for speech recognition, using the PocketSphinx gstreamer module. https://github.com/alumae/ruby-pocketsphinx-server, Abruf: 23.12.2014.
- Grabianowski, E. (o.J.a): How Speech Recognition Works - Page 1. http://electronics.howstuffworks.com/gadgets/high-tech-gadgets/speech-recognition.htm, Abruf: 21.12.2014.
- Grabianowski, E. (o.J.b): Speech Recognition and Statistical Modeling. http://electronics.howstuffworks.com/gadgets/high-tech-gadgets/speech-recognition2.htm, Abruf: 21.12.2014.
- Grabianowski, E. (o.J.c): Speech Recognition Weaknesses and Flaws. http://electronics.howstuffworks.com/gadgets/high-tech-gadgets/speech-recognition3.htm, Abruf: 12.01.2015.
- Grabianowski, E. (o.J.d): Speech to Data. http://electronics.howstuffworks.com/gadgets/high-tech-gadgets/speech-recognition1.htm, Abruf: 21.12.2014.
- Grabianowski, E. (o.J.e): The Future of Speech Recognition. http://electronics.howstuffworks.com/gadgets/high-tech-gadgets/speech-recognition4.htm, Abruf: 21.12.2014.
- Grasch, P. (2013): Open Source Dictation: Acoustic Model. http://grasch.net/node/21, Abruf: 22.12.2014.
- Handermann, S. (2013): Was ist die CPU-Last? CHIP. http://praxistipps.chip.de/was-ist-die-cpu-last_10084, Abruf: 13.12.2014.

- Heise Open Source (o.J.): Support ist kein Problem. http://www.heise.de/open/artikel/Support-ist-kein-Problem-224512.html, Abruf: 07.12.2014.
- htk (o.J.): HTK Extensions. http://htk.eng.cam.ac.uk/extensions/, Abruf: 10.01.2015.
- IBM (o.J.): IBM Desktop ViaVoice. http://www-01.ibm.com/software/pervasive/viavoice.html, Abruf: 03.01.2015.
- Informer Technologies Inc. (o.J.): Anwenderforum für Dragon NaturallySpeaking. http://dragon-spracherkennung.forumprofi.de/, Abruf: 12.12.2014.
- IT Wissen (o.J.a): Back-End. http://www.itwissen.info/definition/lexikon/Back-End-back-end.html, Abruf: 09.12.2014.
- IT Wissen (o.J.b): Spracherkennung. http://www.itwissen.info/definition/lexikon/Spracherkennung-voice-recognition.html, Abruf: 21.12.2014.
- IVONA Software (o.J.): Text-to-Speech. Technologies that create lifelike voices. http://www.ivona.com/us/about-us/text-to-speech/, Abruf: 10.01.2015.
- Jahnke, S. (2006): Digitalisierung von Audio. http://skip-bo.s-jahnke.de/skipbo/Codecs.pdf, Abruf: 10.01.2015.
- Jura Forum (2011): Software-Lizenz. http://www.juraforum.de/lexikon/software-lizenz, Abruf: 09.12.2014.
- Kiecza (o.J.): Introduction. http://www.kiecza.net/daniel/linux/cvoicecontrol/index-1.html, Abruf: 10.01.2015.
- KTH (o.J.): The NICO Toolkit. http://nico.nikkostrom.com/, Abruf: 10.01.2015.
- Lai, S. L. C. / Zhao, Q. (o.J.): Performance Analysis of Speech Recognition Software. http://www.hpcaconf.org/hpca8/sites/caecw-02/s4p1.pdf, Abruf: 13.12.2014.
- Landesakademie für Fortbildung und Personalentwicklung an Schulen (o.J.): Lizenzen für Software. http://lehrerfortbildung-bw.de/sueb/recht/liz/softw/, Abruf: 09.12.2014.

- Lenzo, K. (o.J.): The CMU Pronouncing Dictionary. http://www.speech.cs.cmu.edu/cgi-bin/cmudict, Abruf: 03.01.2015.
- Linguatec (o.J.): Basics of speech recognition. https://docs.google.com/document/d/1teiq6Eb8GcfFirqiHal-7hTE2sdf21g3x0UhIpqY2bI/edit#, Abruf: 06.12.2014.
- Marquard, S. (2011): Sphinx4 speech recognition results for selected lectures from Open Yale Courses. http://trulymadlywordly.blogspot.de/2011/12/sphinx4-speech-recognition-results-for.html, Abruf: 03.01.2015.
- Microsoft (o.J.): Erstellen und Verwenden von Benutzerprofilen zur Spracherkennung. http://office.microsoft.com/de-de/powerpoint-help/erstellen-und-verwenden-von-benutzerprofilen-zur-spracherkennung-HP003084401.aspx, Abruf: 07.12.2014.
- Mims, C. (2014): Intel's voice recognition will blow Siri out of the water—because it doesn't use the cloud. http://qz.com/170668/intels-voice-recognition-will-blow-siri-out-of-the-water-because-it-doesnt-use-the-cloud/, Abruf: 13.12.2014.
- Myers, R. (1996): Myers' Hidden Markov Model software. http://www.speech.cs.cmu.edu/comp.speech/Section6/Recognition/myers.hmm.html, Abruf: 10.01.2015.
- Noyes, K. (o.J.): How to Get Support for Open Source Software. PC World. http://www.pcworld.com/article/207958/how_to_get_support_for_open_source_software.html, Abruf: 11.12.2014.
- Nuance (2011): User Workbook - Dragon Dictate. Speech Recognition Software for the Mac. http://www.nuance.de/ucmprod/groups/dragon/@web-enus/documents/webasset/nc_017363.pdf, Abruf: 17.11.2014.
- Nuance (2014): Dragon NaturallySpeaking 13 Professional. Datenblatt. http://www.nuance.de/ucmprod/groups/dragon/@web-de/documents/collateral/nc_034076.pdf, Abruf: 20.12.2014.

- Nuance (o.J.a): Dragon Dictation. Frequently Asked Questions. http://www.nuancemobilelife.com/support/usa/engusa/dragon-dictation/, Abruf: 13.12.2014.
- Nuance (o.J.b): Dragon Naturally Speaking - Administrator Guide. http://www.nuance.com/ucmprod/groups/corporate/@web-enus/documents/collateral/nc_025639.pdf, Abruf: 12.12.2014.
- Nuance (o.J.c): Dragon Naturally Speaking 12 - Datasheet. http://www.nuance.com/ucmprod/groups/dragon/@web-enus/documents/webasset/nc_008811.pdf, Abruf: 20.12.2014.
- Nuance (o.J.d): Dragon NaturallySpeaking SDK Client Edition. http://www.nuance.com/for-developers/dragon/client-sdk/index.htm, Abruf: 20.12.2014.
- Nuance (o.J.e): Dragon Spracherkennungssoftware. http://www.nuance.de/dragon/index.htm, Abruf: 14.12.2014.
- Nuance (o.J.f): FAQ. http://dragonmobile.nuancemobiledeveloper.com/public/index.php?task=help#support, Abruf: 20.12.2014.
- Nuance (o.J.g): Integrate speech into your app. http://dragonmobile.nuancemobiledeveloper.com/public/index.php?task=prodDev#guide, Abruf: 20.12.2014.
- Nuance (o.J.h): Support - Hardware Compatibility List. http://support.nuance.com/compatibility/Search.asp?PRO=DNS&CID=15, Abruf: 20.12.2014.
- PC Mag (o.J.): Definition of:software integration. http://www.pcmag.com/encyclopedia/term/65245/software-integration, Abruf: 14.12.2014.
- PC Welt (o.J.): Mit Spracherkennung Zeit sparen. Weniger Tippfehler - einfach und bequem. http://www.pcwelt.de/ratgeber/Anzeige-Mit-Spracherkennung-Zeit-sparen-6690136.html, Abruf: 07.12.2014.

- Politepix (o.J.): OpenEars: free speech recognition and speech synthesis for the iPhone. http://www.politepix.com/openears/, Abruf: 10.01.2015.
- Pressrelations (1998): IBM erweitert ihre Spracherkennungsfamilie um Via-Voice Gold. http://presseservice.pressrelations.de/standard/result_main.cfm?aktion=jour_pm&r=2341&quelle=0&pfach=1&n_firmanr_=54&sektor=pm&detail=1, Abruf: 17.01.2015.
- Pscheidt, S. (o.J.): Ein Lernprogramm zur Spracherkennung mit Neuronalen Netzen. With assistance of Prof. Jürgen Sauer. Fachhochschule Regensburg, Abruf: 21.12.2014.
- Puneet, K. (o.J.): Expanding Dictionary Of Acoustic Model. https://puneetk.com/expanding-dictionary-of-acoustic-model, Abruf: 03.01.2015.
- Reichel, U. (2010): Statistische Sprachmodelle. http://www.phonetik.uni-muenchen.de/~reichelu/kurse/stat_lm/script_reichel_stat_lm.pdf, Abruf: 17.12.2014.
- Roos, D. (o.J.): How to Leverage an API for Conferencing. What is an API? http://money.howstuffworks.com/business-communications/how-to-leverage-an-api-for-conferencing1.htm, Abruf: 15.12.2014.
- Roth, W. (2009): IBM Via Voice ist zurück. http://neuerdings.com/2009/04/03/linguatec-voice-pro-12-ibm-via-voice-ist-zurueck/, Abruf: 05.01.2015.
- Rupp, M. (2012): Das richtige Betriebssystem für's Tablet. http://www.pcwelt.de/ratgeber/Das-richtige-Tablet-Betriebssystem-7051388.html, Abruf: 09.12.2014.
- Samtani, G. (2002): Using Web services for application integration. http://www.techrepublic.com/article/using-web-services-for-application-integration/, Abruf: 14.12.2014.
- Shanklin, W. (2013): 2013 Tablet Comparison Guide. http://www.gizmag.com/tablet-comparison-2013/29708/, Abruf: 13.12.2014.

- Simon listens (2012): Simon. http://www.simon-listens.org/index.php?id=122&L=1, Abruf: 10.01.2015.
- Sourceforge (2009): Sphinx4 as a Java Webservice. http://sourceforge.net/p/cmusphinx/discussion/sphinx4/thread/c622ed1c/, Abruf: 21.12.2014.
- Sourceforge (2014a): How to extend LM to 5-grams in cmuclmtk? http://sourceforge.net/p/cmusphinx/discussion/help/thread/a3174a5f/?limit=25, Abruf: 20.12.2014.
- Sourceforge (2014b): PocketSphinx API Documentation. http://cmusphinx.sourceforge.net/doc/pocketsphinx/, Abruf: 20.12.2014.
- Sourceforge (2015): Kaldi. http://kaldi.sourceforge.net/about.html, Abruf: 10.01.2015.
- Sourceforge (o.J.a): Class LargeTrigramModel. http://www.gavo.t.u-tokyo.ac.jp/~kuenishi/java/sphinx4/edu/cmu/sphinx/linguist/language/ngram/large/LargeTrigramModel.html, Abruf: 03.01.2015.
- Sourceforge (o.J.b): Class Microphone. http://cmusphinx.sourceforge.net/doc/sphinx4/edu/cmu/sphinx/frontend/util/Microphone.html, Abruf: 20.12.2014.
- Sourceforge (o.J.c): Class Utterance. http://cmusphinx.sourceforge.net/doc/sphinx4/edu/cmu/sphinx/frontend/util/Utterance.html, Abruf: 20.12.2014.
- Sourceforge (o.J.d): CMU Sphinx Description. http://sourceforge.net/projects/cmusphinx/, Abruf: 20.12.2014.
- Sourceforge (o.J.e): CMU Sphinx Forum. http://sourceforge.net/p/cmusphinx/discussion/, Abruf: 20.12.2014.
- Sourceforge (o.J.f): eSpeak text to speech. http://espeak.sourceforge.net/, Abruf: 10.01.2015.
- Sourceforge (o.J.g): XVoice. http://sourceforge.net/p/xvoice/mailman/, Abruf: 03.01.2015.

- SpeechRecognition Solutions (2013): Dragon Support and Troubleshooting Tutorials. http://www.speechrecsolutions.com/tutorials.html, Abruf: 10.01.2015.
- SpeechRecognition Solutions (o.J.): Microphone Selection Guide: USB Adapters. http://www.speechrecsolutions.com/mic_guide_usbadapters.html, Abruf: 07.01.2015.
- Spehr, M. (2014): Spracherkennung Dragon 13. Endlich versteht uns der Computer perfekt. FAZ. http://www.faz.net/aktuell/technik-motor/computer-internet/gesagt-getan-sprachprogramm-dragon-13-im-praxistest-13128598.html, Abruf: 20.12.2014.
- sphinxkeys (2011): Wiki. https://code.google.com/p/sphinxkeys/wiki/README, Abruf: 10.01.2015.
- Sundaram, R. et al. (o.J.): ISIP 2000 CONVERSATIONAL SPEECH EVALUATION SYSTEM. Mississippi State University. http://www.isip.piconepress.com/publications/conference_proceedings/2000/dod_stw/asr/doc/paper_final.pdf, Abruf: 10.01.2015.
- TLDP (o.J.): Speech Recognition Software. http://www.tldp.org/HOWTO/Speech-Recognition-HOWTO/software.html, Abruf: 10.01.2015.
- TU Berlin (2011): Linguistik. http://www.linguistik.tu-berlin.de/, Abruf: 21.12.2014.
- Universität Aachen (2014): RWTH ASR - The RWTH Aachen University Speech Recognition System. http://www-i6.informatik.rwth-aachen.de/rwth-asr/, Abruf: 10.01.2015.
- Universität Bonn (o.J.): N-Gramme. http://www.sk.uni-bonn.de/lehre/informationen-materialien/informationen-und-materialien-kopho/materialien-1/breuer/transkription-in-sprachsynthese-und-erkennung/N-Gramme.pdf, Abruf: 13.12.2014.
- Universität Oldenburg (o.J.): Immatrialgüterrecht. http://www.informatik.uni-oldenburg.de/~iug09/igr/koenigssee.informatik.uni-

oldenburg.de_1211/index.php/software/23-lizenzen/78-formen-von-softwarelizenzen.html, Abruf: 09.12.2014.

- Universität Ulm (1996): Jialong He's Speech Recognition Research Tool. http://www.speech.cs.cmu.edu/comp.speech/Section6/Recognition/jialong.html, Abruf: 10.01.2015.
- University Dortmund (o.J.): Theorie: Standortplanung. Methoden zur Festlegung der Gewichtung einzelner Kriterien (z.B. paarweiser Vergleich o. Bewertungsmatrix). https://ews.tu-dortmund.de/public/lecture/logedugate/public/LFO/standortplanung/content/08_gewichtung.htm, Abruf: 07.12.2014.
- Voicerecog (2006): Bluetooth Headset results. http://www.xncroft.com/blog/lyceum/voicerecog/2006/06/30/bluetooth-headset-results/, Abruf: 21.12.2014.
- Wheeler, D.A. (2011): How to Evaluate Open Source Software / Free Software (OSS/FS) Programs. http://www.dwheeler.com/oss_fs_eval.html, Abruf: 29.11.2014.
- Wikipedia (2014a): Emacspeak. http://en.wikipedia.org/wiki/Emacspeak, Abruf: 10.01.2015.
- Wikipedia (2014b): Festival Speech Synthesis System. http://en.wikipedia.org/wiki/Festival_Speech_Synthesis_System, Abruf: 10.01.2015.
- Wikipedia (2014c): Julius (software). http://en.wikipedia.org/wiki/Julius_%28software%29, Abruf: 10.01.2015.
- Wikipedia (2014d): Spracherkennung. http://de.wikipedia.org/wiki/Spracherkennung#Formanten, Abruf: 09.12.2014.
- Wikipedia (2015): Dragon Naturally Speaking. http://en.wikipedia.org/wiki/Dragon_NaturallySpeaking, Abruf: 14.01.2014.
- Wikipedia (o.J.): Modular Audio Recognition Framework; http://marf.sourceforge.net/.

http://en.wikipedia.org/wiki/Modular_Audio_Recognition_Framework;_http://marf.sourceforge.net/, Abruf: 10.01.2015.

- Windows (o.J.): GVoice. http://apps.microsoft.com/windows/en-us/app/gvoice/b34a71c0-c183-432c-8780-64d0e2466a16, Abruf: 10.01.2015.



Bewertung und Vergleich von Open Source und kommerziellen Tools zur Unterstützung leichtgewichtiger Softwareentwicklung

(Scrum & Kanban)

Schriftliche Ausarbeitung
im Rahmen der Lehrveranstaltung „Integrationsseminar“
für das Kompetenzzentrum Open Source (KOS)

vorgelegt von

Eva Kuron
Stefanie Pusch
Lisa-Marie Orzelski

am 26.01.2015

Fakultät Wirtschaft
Studiengang Wirtschaftsinformatik
WWI2012I

Inhaltsverzeichnis

Abkürzungsverzeichnis

ALM	Application Life Cycle Management
EL	Effort Left
OE	Original Estimate
OS	Open Source
SaaS	Software as a Service
WIP	Work in Progress

Abbildungsverzeichnis

1 Einleitung

Die leichtgewichtige Softwareentwicklung löst nach und nach die klassischen, schwergewichtigen Methoden der Entwicklung von Software ab. Erhöhte Flexibilität, bessere Qualität und geringere Risiken sind nur einige Gründe sich für den leichtgewichtigen oder agilen Weg zu entscheiden. Es gibt zahlreiche Tools mit verschiedensten Funktionalitäten zur Unterstützung der neuen Methoden, sodass es zum Teil schwer fallen kann sich für das richtige Werkzeug zu entscheiden. Um diese Produktauswahl zu erleichtern, wird im Rahmen dieser Arbeit ein Kriterienkatalog erstellt. Anhand dieser Matrix werden die Tools zur leichtgewichtigen Softwareentwicklung fundiert bewertet.

Ziel der Arbeit ist es einen Vergleich zu den momentan am Markt befindlichen leichtgewichtigen Tools im kommerziellen und Open Source (OS) Bereich durchzuführen und eine Bewertungsgrundlage zur Auswahl eines auf die Anforderungen des jeweiligen Unternehmens passenden Tools zu schaffen.

Nach den einleitenden Kapiteln folgt eine Einführung in die leichtgewichtige Softwareentwicklung mit besonderem Fokus auf Scrum und Kanban. Das dritte Kapitel erläutert den im Projekt entworfenen Kriterienkatalog mit dessen Kriterien. Des Weiteren wird in diesem Kapitel die Anwendungsweise des Kriterienkatalogs geschildert. Im darauf folgenden Kapitel wird eine detaillierte Analyse der ausgewählten Tools im kommerziellen sowie im Open Source Bereich durchgeführt. Die Arbeit wird mit einer Tool-Empfehlung im Fazit abgeschlossen.

2 Theoretische Grundlagen

2.1. Leichtgewichtige Softwareentwicklung

Zur Softwareentwicklung stehen diverse Vorgehensmodelle zur Auswahl. Grundsätzlich lassen sich diese Modelle in zwei Kategorien unterteilen, den „klassischen“, sowie den „agilen“ Ansatz zur Entwicklung von Software.[1]

Klassische Softwareentwicklung

Bei der „klassischen“ Vorgehensweise liegt dabei der Fokus auf einem formalen Prozess (vgl. Abb. 1), welcher als statisch und unflexibel angesehen wird. So wird während der Planungsphase ein detaillierter Prozess zur Entwicklung der Software festgelegt.[2] Zu den wichtigsten Vertretern der klassischen Softwareentwicklung zählen das Wasserfallmodell und das Spiralmodell.



Abb 1: Plan-driven vs. vision-driven[3]

[1] Vgl. Hense, A. (2012), Abruf: 05.12.2014
[2] Vgl. Tan, E./Weiss, J. (2004), Abruf: 05.12.2014
[3] Enthalten in: Bock, J. (2013), Abruf: 12.12.2014

Entstehung der agilen Softwareentwicklung

Im Gegensatz dazu konzentriert sich der „agile“ Ansatz auf die Kommunikation zwischen dem Entwickler und dem Kunden. Diese Art der Softwareentwicklung wird auch als „leichtgewichtig“ bezeichnet und entstand in den 1990er Jahren.[4] Der Begriff „agil“ wurde jedoch erst 2001 erstmals im „Manifest der agilen Softwareentwicklung“ durch 12 essentielle Prinzipien definiert:[5]

1. Den Kunden zufrieden zu stellen hat höchste Priorität
2. Änderungen zu Anforderungen der Software sind zu jeden Zeitpunkt in der Entwicklung möglich und gewünscht
3. Die Software wird dem Kunden in allen Entwicklungsphasen regelmäßig vorgestellt
4. Tägliche Zusammenarbeit der Entwickler
5. Motivation, Unterstützung und Vertrauen sind essentielle Bestandteile der Zusammenarbeit
6. Face-to-face Gespräche als effizienteste und effektivste Kommunikationsart
7. Der Projektstatus wird an funktionierender Software gemessen
8. Gleichmäßiges Tempo in allen Projektphasen
9. „Technische Exzellenz und gutes Design fördert die Agilität“
10. Mit möglichst wenig Arbeitsaufwand das bestmögliche Ergebnis erreichen (Effizienz)
11. Selbstorganisierte Teams
12. Regelmäßige Teammeetings zur Reflexion und Anpassung

Erfolgspotenzial des agilen Ansatzes

RESOLUTION

	2002	2004	2006	2008	2010
Successful	34%	29%	35%	32%	37%
Challenged	51%	53%	46%	44%	42%
Failed	15%	18%	19%	24%	21%

Project resolution results from CHAOS Research for years 2002 to 2010.

Abb 2: Erfolgsrate von Projekten[6]

[4] Vgl. Tan, E./Weiss, J. (2004), Abruf: 05.12.2014
[5] Vgl. Beck u. a. (2001), Abruf: 01.12.2014
[6] Enthalten in: The Standish Group International, Inc (2011), Abruf: 05.12.2014

In 2002 verfolgten noch weniger als 5% der Softwareprojekte den agilen Ansatz. In 2012 stieg der Anteil bereits auf 29% an (vgl. Abb. 2).[7] Laut Chaos Manifesto steht die Erfolgsrate der Projekte in direktem Zusammenhang mit der verwendeten Methode. Das Scheitern vieler Projekte mit klassischem Ansatz liegt oftmals in Fehlplanungen begründet. Es wurden Zeitaufwand und Budget fehlerhaft kalkuliert und Kundenbedürfnisse sowie Marktvoraussetzungen ungenügend betrachtet. Folgen seien dann vermehrter Stress, unzufriedene Kunden, sowie ein Endprodukt mit schlechter Wirtschaftlichkeit.[8]

2.2. Scrum

Unter Scrum versteht man ein agiles und auf den Menschen ausgerichtetes Framework für das Projektmanagement und keinen vordefinierten Prozess, bei dem man sich an eine bestimmte Abfolge von Schritten halten muss. Der Fokus von Scrum liegt am Ende auf dem funktionierenden Endergebnis bzw. Produkt.[9] Mit Hilfe von Scrum läuft die Produktentwicklung iterativ-inkrementell ab, d.h. dass nach jedem Sprint ein fertiges Produktstück des Gesamtproduktes fertiggestellt sein muss.[10] Unter einem Sprint versteht man Iterationen während des Projektes.

Ein Scrum-Projekt wird von drei wesentlichen Rollen umgesetzt: dem Product Owner, dem Scrum Master und dem Team. Als vierte Rolle könnte man noch die Stakeholders hinzuzählen, da auch diese das Projekt beeinflussen. Die verschiedenen Rollen haben jeweils unterschiedliche Aufgaben.

Der Product Owner kann als der Auftraggeber gesehen werden. Er stellt die fachlichen Anforderungen und ordnet sie nach ihrer Priorität. Seine Ergebnisse gibt er anschließend an das Team und den Scrum Master weiter. Trotzdem ist er nicht der Chef des Scrum-Teams. Die Anforderungen werden im sogenannten Product Backlog gepflegt. Das Produkt Backlog lässt sich mit einer Liste vergleichen. Diese Liste muss andauernd überarbeitet und angepasst werden. Der Product Owner erstellt, unter Berücksichtigung des höchsten Ziels, kleine Arbeitspakete, die vom Team umgesetzt werden sollen. Ein Arbeitspaket wird als Increment bezeichnet und wird während eines Sprints nicht mehr bearbeitet, sondern es werden nur die zu Beginn gesetzten Anforderungen beachtet, damit keine Fehler bei der Fertigstellung des Produktes entstehen. Viele der Arbeitspakete werden in noch kleinere Arbeitspakete unterteilt, sogenannte Tasks.

[7] Vgl. The Standish Group International, Inc (2011), Abruf: 05.12.2014
[8] Vgl. Bock, J. (2013), Abruf: 12.12.2014
[9] Vgl. Rubin, K. (2014), S. 47
[10] Vgl. Dräther, R./Koschek, H./Sahling, C. (2013), S. 13f.

Jeder Task besitzt ebenfalls einen eigenen Bearbeiter und der Fortschritt wird im Sprint Backlog festgehalten.[11]

Der Scrum Master hingegen überwacht den Prozess und kontrolliert, ob alle Regeln eingehalten werden und die Teammitglieder ihren aktuellen Status täglich im Sprint Backlog aktualisieren. Außerdem besteht die Aufgabe des Scrum Masters in der Unterstützung des Informationsaustausches zwischen dem Team und dem Product Owner.

Das Scrum-Team ist zuständig für die tatsächliche Entwicklung des Produkts, d.h. für die Umsetzung der ihnen gestellten Anforderungen des Product Owners. Ein Team besteht mindestens aus fünf und maximal aus zehn Mitgliedern. Sollte eine Gruppe doch größer sein, wird sie in kleinere Teams aufgeteilt. Dem Team werden die Anforderungen vorgegeben, jedoch nicht das Vorgehen bzw. die Entscheidungen, die getroffen werden müssen, um den Anforderungen nachzukommen, d.h. dass das Team eigenen Entscheidungsspielraum hat und ohne Projektleiter zurechtkommt.

Zu Beginn wurden Sprints bereits erwähnt. Sprints spielen eine wesentliche Rolle in einem Scrum-Projekt. Während eines Sprints werden die Tasks vom Team bearbeitet und es wird den Anforderungen nachgekommen, da am Ende eines Sprints ein fertiges und einsatzfähiges Stück des Gesamtproduktes geliefert werden muss. In der Regel beträgt die Länge eines Sprints ca. 30 Tage. Innerhalb eines Scrum-Projektes finden alle Meetings und Termine in vordefinierten Zeitabschnitten statt, die nicht überschritten werden sollten. Diese vordefinierten Zeitabschnitte nennt man Time-Boxes. Neben den Zeitabschnitten wird i.d.R. auch die Länge bzw. Dauer eines Meetings vorher bereits festgelegt, z.B. geht ein Daily Scrum Meeting ca. 15 Minuten. Ein Daily Scrum Meeting dient dazu den aktuellen Stand eines Projektes zu überprüfen. Während des Meetings wird der Status jedes einzelnen Teammitglieds besprochen, damit jeder im Team weiß, wo die Anderen stehen. Dieses Meeting wird auch genutzt um aufgetretene Probleme und Schwierigkeiten innerhalb eines Tasks zu besprechen. Die Teammitglieder können sich so gegenseitig weiterhelfen und die Fertigstellung des Prozesses vorantreiben.

Ebenfalls zu Beginn und Ende eines Sprints finden Meetings statt. Am Anfang findet das Sprint Planning Meeting statt. Dieses Meeting dauert ca. acht Stunden und hier werden die Arbeitspakete und Anforderungen zunächst definiert und entwickelt. In der Zwischenzeit werden die Anforderungen nicht mehr geändert, da das Team ungestörte Entwicklungszyklen benötigt, um das gesetzte Ziel erreichen zu können.[12] Am Ende des Sprints findet das Sprint Review Meeting

[11]Vgl. Dräther R./Koschek, H./Sahling, C. (2013), S. 14
[12] Vgl. it-agile GmbH (o. J.), Abruf: 06.12.2014

statt. Während dieses Meetings präsentiert das Scrum-Team das fertige Produktstück. An diesen Meetings nehmen der Product Owner und gegebenenfalls auch die Stakeholder und weitere Interessierte teil. Die Planung für den nächsten Sprint ergibt sich aus dem Feedback und den Anforderungen des Publikums.

Der Scrum Master ist ebenfalls für die Burndown-Charts verantwortlich. Diese Charts dienen der Visualisierung des Fortschritts des aktuellen Sprints und des gesamten Projekts. Hiermit können Probleme schnell erkannt und gegebenenfalls auch vermieden werden.[13]

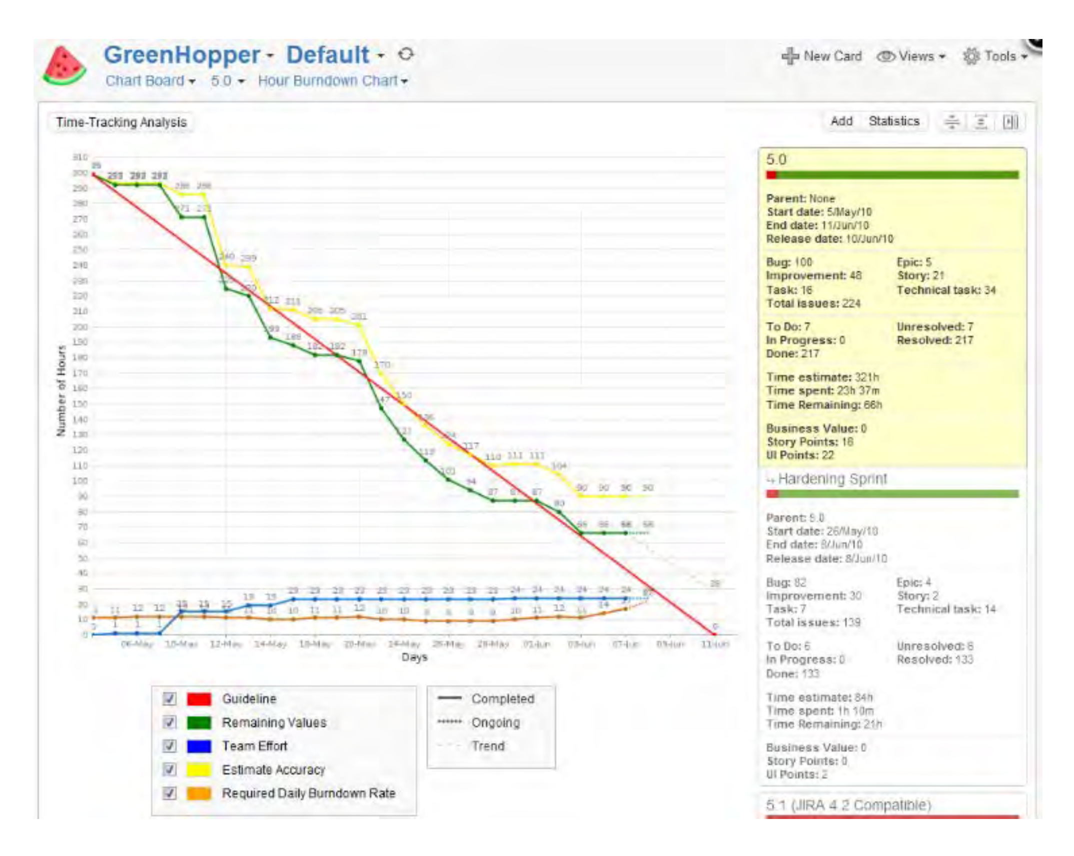


Abb 3: Hour Burndown Chart (Atlassian) [14]

Die obere Abbildung (Abb. 3) zeigt ein „Hour Burndown Chart" aus dem JIRA Agile Tool von Atlassian. Die rote Linie wird als Leitlinie gesehen, da sie den idealen Burndown beschreibt, der sich aus Schätzungen und gesammelten Erfahrungen der vorigen Sprints ergibt. Die anderen Linien geben Informationen über die Aufwendungen des Teams, die erforderliche tägliche Burndown-Rate und noch ausstehende Anforderungen und Aufgaben.

[13] Vgl. Scrum-master.de (2014), Abruf: 06.12.2014
[14] Enthalten in: Atlassian Inc. (2015c), Abruf: 01.12.2014

Es gibt drei essentielle Voraussetzungen an ein Scrum-Projekt. Alle Informationen sollten immer verfügbar und für jedermann im Team sichtbar sein, d.h. Transparenz ist erwünscht. Zweitens sollte die eigene Vorgehensweise regelmäßig überprüft werden. Die dritte Voraussetzung stellt die Anpassung der eigenen Vorgehensweisen dar, damit das Ziel besser und effizienter erreicht werden kann.[15]

2.3. Kanban

Kanban ist eine Methode, die zur agilen Softwareentwicklung genutzt wird und sich auf die Optimierung des Workflows und die fließende Abarbeitung der einzelnen Aufgaben konzentriert. Mithilfe von Kanban sind Teams in der Lage kontinuierlich Ergebnisse zu liefern. Wichtig dabei ist auch, dass diese Methode es durch das Kanban-Board möglich macht, Aufgaben visuell ansprechender und übersichtlicher darzustellen.[16]

Kanban kommt aus dem japanischen und bedeutet „Signalkarte“.[17] Es wurde erstmals bei Toyota in Zusammenhang mit der „Just-in-time“ Produktion und dem „Lean Management“ im Produktionsumfeld angewendet. In der IT- und Software-Industrie wird Kanban als Management-Methode verwendet, um schrittweise die Lieferung von Ergebnissen (z.B. Software, Services) zu optimieren.[18] Es wurde in diesem Sinne erstmals im Jahr 2007 von David J. Anderson, dem „Vater des Kanban für Softwareentwicklung“, der Öffentlichkeit vorgestellt.[19] Es gibt verschiedene Zusammenfassungen der Grundregeln des Kanban in der Fachliteratur, jedoch beziehen sich alle auf dieselben grundlegenden Prinzipien. Es lässt sich zusammenfassen, dass es drei bedeutende Kanban Prinzipien gibt:[20]

1. Beginne mit dem aktuellen Prozess, dem Ist-Zustand
2. Bringe Veränderungen und Verbesserungen ein
3. Respektiere die vorhandenen Rollen und Stellungen im Unternehmen

Auf diese drei Grundannahmen bauen die sechs Schritte für die erfolgreiche Anwendung des Kanban auf:

1. Visualisiere die Arbeit und den Ablauf

[15] Vgl. Dräther R./Koschek, H./Sahling, C. (2013), S. 14f.
[16] Vgl. Epping, T. (2011), S. 24ff.
[17] Vgl. Roock, A. (o. J.), Abruf: 20.12.2014
[18] Vgl. Quack, K.(2014), Abruf: 05.12.2014
[19] Vgl. Epping, T.(2011), S. 1
[20] Vgl. Digité, Inc. (2015b), Abruf: 02.12.2014

2. Begrenze die Menge paralleler Arbeiten (Work in Progress (WIP))
3. Steuere den Workflow
4. Stelle klare Regeln auf
5. Nutze vorhandene Modelle
6. Implementiere Feedback-Kreisläufe

Das korrekt angewandte Kanban hilft dem Team enger zusammenzuarbeiten und den aktuellen Prozess gemeinsam zu verbessern. Im Gegensatz zu anderen Vorgehensweisen kann bei Kanban der Prozess selbst entworfen und angepasst werden. Es geht dabei darum, die Qualität der Ergebnisse zu erhöhen und den Durchlauf zu steigern. Da Kanban sehr einfach einzuführen und anzuwenden ist, wird es immer populärer in verschiedenen Unternehmensbereichen.

Entsprechend der Wortbedeutung werden beim Kanban-System Karten auf einer virtuellen Tafel (Kanban-Board) angeordnet, um den Status der einzelnen Aufgaben anzuzeigen.[21] Es baut auf der traditionellen Arbeit mit klassischen Plakaten, Flipcharts und Notizzetteln auf. Spalten, welche die einzelnen Arbeitsschritte wie zum Beispiel „Test“ visualisieren, zeigen den Platz der Karten im Entwicklungsprozess an. Die Spalten haben dabei ein Limit von Karten, die sich im jeweiligen Bereich befinden können (Work in Progress Limit). Ist in einer Spalte Raum für eine neue Karte, das heißt es gibt freie Arbeitskraft, wird mittels des Pull-Prinzips eine der fertigen Aufgabenkarten aus der vorherigen Spalte oder dem Backlog gezogen. Jede Karte auf der Tafel kann und sollte wieder zusätzliche Informationen enthalten, zum Beispiel bezüglich des verantwortlichen Teammitglieds.

Kanban betont Verantwortlichkeit und die Zuordnung einzelner Personen zu „ihren“ Aufgaben. Allerdings sollten Aufgaben erst dann fest zugeordnet werden, wenn sie auch tatsächlich in Bearbeitung sind. Wichtig ist außerdem, dass Kanban es dem Team ermöglicht, Hindernisse und Engpässe schneller zu erkennen und zu optimieren. Dazu sollen parallele Arbeiten in den einzelnen Spalten limitiert werden.[22] Abbildung 4 verdeutlicht ein Hindernis bei den Testern, welches es durch die Begrenzung der Karten/Aufgaben den Entwicklern versperrt, neue Arbeit hinzuzufügen.

[21] Vgl. Roock (o. J.), Abruf: 20.12.2014
[22] Vgl. Ladas, C. (2008), S. 45ff.

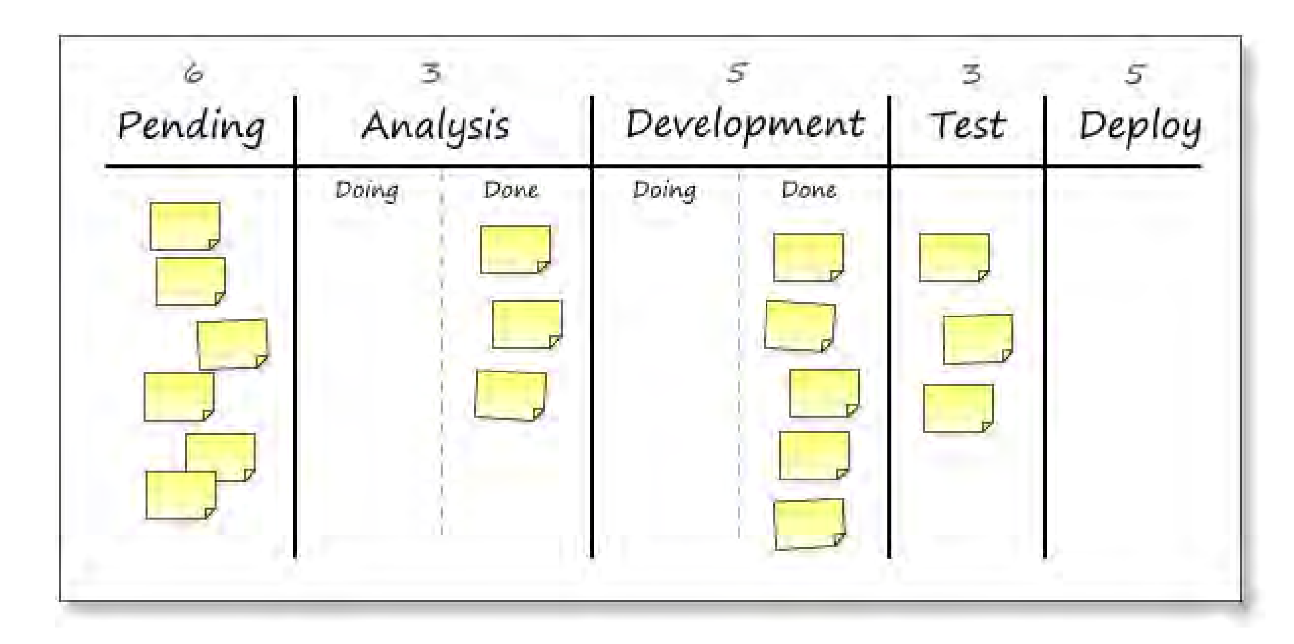


Abb 4: Kanban Board mit WIP-Limit[23]

Der Prozess muss im Falle eines solchen Engpasses angepasst werden, um die Durchlaufzeiten zu verbessern, dabei sollte gezielt das vorhandene Hindernis adressiert werden. Durch online Kanban-Anwendungen wird es dem Team vereinfacht, an verschiedenen Orten verbunden zu bleiben und zu kommunizieren.

Beim Kanban werden zusätzlich verschiedene Kennzahlen gesammelt.[24] Das Cumulative Flow Diagram macht deutlich, wann wie viele Tasks erledigt sein sollen und zeigt deren Zustände im Kanban-Workflow. Das Work in Progress zeigt an, wie sich die Anzahl paralleler Arbeiten mit der Zeit entwickelt. Der Durchsatz (Throughput) gibt an, wie viele Tickets bzw. Aufgaben pro Woche erledigt wurden. Zuletzt zeigt die Fehlerrate die Anzahl der Fehler mit der Zeit an.

Bei der Auswahl einer Anwendung zur Unterstützung der Arbeit mit Kanban ist es wichtig auf einige Faktoren zu achten. Es ist wesentlich zu planen, wie viele Nutzer die Software nutzen werden und wie viele Projekte damit bearbeitet werden sollen. Außerdem sollte bedacht werden, ob das Tool allein oder in Zusammenarbeit mit anderer Software (z.B. Scrum) genutzt werden soll. Die Software sollte dazu einige grundlegende Eigenschaften vorweisen, wie etwa ein Limit für parallel laufende Arbeiten (WIP). Kanban Karten sollten frei anpassbar sein und die Möglichkeit bieten, mehreren Teammitgliedern zugeordnet zu werden. Je nachdem, ob mit anderer Software zusammen gearbeitet werden soll, sollte das Tool Integrationsmöglichkeiten bieten.

[23] Enthalten in: Peterson, D. (2014), Abruf: 01.12.2014
[24] Vgl. Ebenda, Abruf: 01.12.2014

Zudem ist es hilfreich, wenn verschiedene Metriken berechnet werden, um den Arbeitsfluss zu analysieren.

Werden Kanban und Scrum zusammen zur leichtgewichtigen Softwareentwicklung genutzt, so nennt man diese Kombination auch „Scrumban“. Hier wird das agile Framework Scrum durch die Methodik des Kanban unterstützt.[25] Beide haben einige Gemeinsamkeiten, etwa gibt es tägliche Feedback-Meetings beim Kanban, welche den Scrum-Meetings (Daily Scrum) ähneln. Scrum bietet die Vorteile, dass schnell auf Veränderungen reagiert werden kann und iterative Tests durchgeführt werden. Allerdings muss auch auf die Unterschiede zwischen dem agilen Scrum-Framework und der leichtgewichtigen Kanban-Methode geachtet werden. Kanban basiert mehr auf dem Kontroll-prinzip während Scrum mehr auf Menschen und Kollaboration fokussiert ist.[26] Kanban kann in diesem Sinn in bestimmten Schritten des Scrum-Kreislaufs eingesetzt werden, um die Fehler bei der Entwicklung zu minimieren und den Durchlauf der einzelnen Arbeitsschritte zu steigern. Gemeinsam bilden Scrum und Kanban ein starkes Team, wenn das Projekt-Team weiß, wie man sie gezielt einsetzt.

2.4. Markt der leichtgewichtigen Tools

Eine Betrachtung des Marktes für Tools zur Unterstützung leichtgewichtiger Softwareentwicklung zeigt eine Vielfalt von verfügbaren Produkten.[27] Das große Angebot spiegelt die wachsende Nachfrage nach agilen Entwicklungsmethoden, wie Scrum und Kanban, welche immer mehr Anwender finden, wider. Gartner unterstreicht zum Beispiel, dass bezüglich des Application Life Cycle Management (ALM) Markts, der Marktanteil, welcher agiles Management betrifft, ein besonderes Wachstum vorweist.[28] Die Marktstudien großer Firmen wie Forrester und Gartner beziehen sich auf ALM Tools, beziehen dabei jedoch agile Vorgehensweisen als einen von drei zentralen Trends mit ein. Zusätzlich zu „Enterprise Agile“ werden „Cloud Computing“ und „Mobile Web“ als Trends erkannt, diese sind für diese Arbeit von weniger Bedeutung. Bei der Untersuchung der Anwendungen muss zwischen kommerziellen Tools auf der einen und Open Source Tools auf der anderen Seite unterschieden werden. Viele große Software- bzw. IT-Firmen bieten kommerzielle, kostenpflichtige Anwendungen an. Hierzu zählen etwa der Agile Manager von Hewlett-Packard oder das Agile Development Tool von IBM. Allerdings fällt auf,

[25] Vgl. Ladas, C.(2008), S. 11f.
[26] Vgl. Sahota, M. (2012), S. 9ff.
[27] Vgl. Mountain Goat Software, LLC (o. J.), Abruf: 20.01.2015
[28] Vgl. Duggan, J./Murphy, T. E. (2012), S. 1

dass der Großteil der Tools auf dem Markt von kleineren Unternehmen angeboten wird. Diese no-name Tools haben gegenüber den größeren Anbietern einen erheblich kleineren Marktanteil, da ihre Produkte weniger bekannt sind und im Vergleich zu Größen wie Rally weniger durch Marketing unterstützt werden. Dies wird zum Beispiel durch eine Studie von Forrester zu Application Life Cycle Management Tools deutlich. Wie die Abbildung (Abb. 5) zeigt, liegen IBM, Rally, HP, Microsoft und CollabNet bei der Marktpräsenz deutlich vorn.

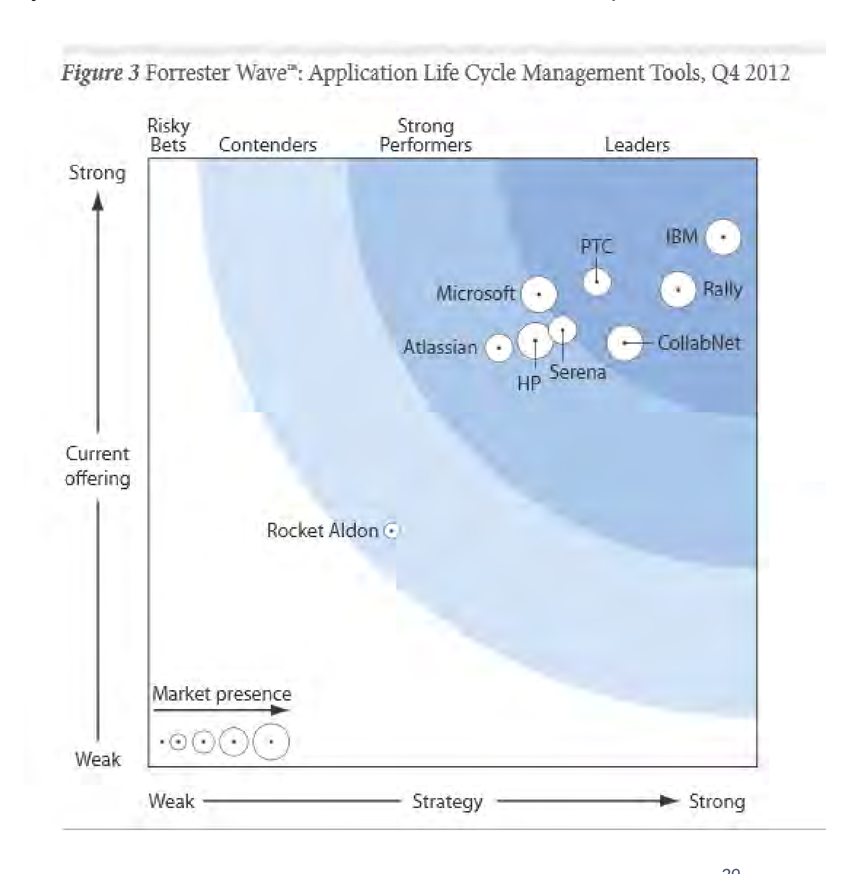


Abb 5: Forrester Wave: Application Life Cycle Management Tools[29]

Auch eine Marktanalyse durch Gartner, die das berühmte "Magic Quadrant" zum Ergebnis hat, unterstreicht die Dominanz weniger großer Firmen.[30] Microsoft, IBM, Atlassian und Rally stehen als „Leaders" an der Spitze des Markts (vgl. Abb. 6). Es fällt auf, dass beide Studien dieselben weitverbreiteten Marktführer aufführen. Die Frage ist allerdings, ob diese Anbieter auch die besten Tools zur leichtgewichtigen Softwareentwicklung zur Verfügung stellen. Diese Frage wird im weiteren Verlauf durch den Vergleich der Tools beantwortet.

[29] Enthalten in: Grant, T. (2012), Abruf: 07.01.2015
[30] Vgl. Duggan, J./Murphy, T.E. (2012), Abruf: 05.01.2015



Abb 6: Gartner Magic Quadrant Application Life Cycle Management 2013[31]

Für den gezielten Vergleich verschiedener Tools muss also eine Vorauswahl aus dem Marktangebot getroffen werden. Für die folgende Bewertung wurden sechs kommerzielle und drei Open Source Tools ausgewählt, welche eine gute Präsenz auf dem Markt haben. Alle Anwendungen werden im Verlauf dieser Arbeit genauer beschrieben und bewertet. Dabei werden gezielt die Anwendungen der etablierten Softwarefirmen denen kleinerer Anbieter gegenübergestellt. Untersucht wird auch, wie frei verfügbare Open Source Tools im Gegensatz zu kostenpflichtigen Angeboten abschneiden und ob diese eine gute Alternative darstellen. Informationen über die einzelnen Tools und deren Nutzung wurden durch das Durchspielen von Testversionen gesammelt. Außerdem stellen alle Anbieter der Tools Details über die Funktionen auf der jeweiligen Website zur Verfügung.

[31] Enthalten in: Atlassian Inc. (2015b), Abruf: 07.01.2015

3 Entwurf eines Kriterienkatalogs

3.1. Auswahl der Kriterien

Um die Bewertung und den Vergleich der ausgewählten Tools zu vereinheitlichen, muss zuerst ein geeigneter Kriterienkatalog erstellt werden. Wie bereits vorhandene Vergleichsstudien von Tools, etwa von Firmen wie Gartner oder Forrester[32] zeigen, ist eine Voraussetzung für ein geeignetes Ergebnis eine einheitliche Basis des Vergleichs. Die ausgesuchten Kriterien ermöglichen eine Bewertung eines jeden Tools nach denselben Gesichtspunkten. Die Kriterien fokussieren sich auf verschiedene Faktoren des Tools, welche anhand von Kategorien deutlich werden.

Die Auswahl der Bewertungskriterien erfolgt auf Basis einer breiten Recherche über notwendige Funktionen und Eigenschaften der Anwendungen.[33] Bei der Recherche wird zuerst eine ausführliche Liste möglicher Kriterien zusammengestellt und hinsichtlich der Schwerpunkte des Vergleichs analysiert. Ein gezieltes Auswahlverfahren basierend auf den Anforderungen von Teams, die leichtgewichtige Methoden verwenden, führt dann zum finalen Kriterienkatalog. Dieser Katalog enthält nur wesentliche Aspekte, welche dabei helfen herauszuarbeiten, wie erfolgreich das Tool die leichtgewichtige Softwareentwicklung unterstützt. Bei der Bewertung geht es darum, zu erkennen, ob eine bestimmte Eigenschaft vorhanden ist und in welcher Ausprägung das Tool diese vorweist. Die Recherche für die Bewertung leichtgewichtiger Tools ergab einerseits funktionale Kriterien der Kategorien Scrum, Kanban und Funktionen der Kollaboration sowie nicht-funktionale Aspekte bezüglich der Implementierung, Schnittstellen, Plattform und Customization. Im Mittelpunkt stehen die funktionalen Aspekte, da mithilfe dieser bewertet wird, wie gut das Tool die Frameworks der leichtgewichtigen Softwareentwicklung verinnerlicht. Basierend auf der gesamten Bewertung eines jeden Tools kann schließlich ein Vergleich nach allen oder einzelnen Faktoren stattfinden. Im Folgenden werden die Kriterien nach den einzelnen Kategorien näher erläutert.

3.1.1. Scrum Kriterien

Für die Kriteriengruppe „Scrum" werden im weiteren Verlauf die wichtigsten Kriterien bzw. Funktionalitäten des agilen Frameworks beschrieben.

[32] Vgl. Grant, T. (2012) Abruf: 07.01.2014
[33] Vgl. Erickson , K.(2013), Abruf: 10.12.2014; Vgl. auch Holdorf, C.(2011), Abruf: 17.01.2015

Wie bereits erwähnt werden zu Beginn eines Projekts verschiedene Anforderungen gestellt. Diese und die sich daraus ergebenen Aufgaben lassen sich in einer Liste, dem sogenannten „Backlog", speichern. Im Kriterienkatalog wird bewertet, wie einfach ein Backlog vom Benutzer erstellt werden kann. Ein Backlog setzt sich aus mehreren „Backlog-Items" zusammen. Ist die Anordnung der Backlog-Items übersichtlich und ist die Anzahl der möglichen Items begrenzt? Diese Anforderungen werden mit dem Punkt „Backlog-Items" bewertet.

Damit die Einträge im Backlog für jeden im Team zugänglich sind, wird in der Regel ein „Task Board" verwendet. Wie intuitiv und übersichtlich dieses gestaltet ist und inwiefern das Task Board den aktuellen Bearbeitungszustand eines Tasks visualisiert, wird unter dem Punkt „Task Board" bewertet.

Die Burndown-Charts dienen der Visualisierung des Fortschritts des aktuellen Sprints oder Releases sowie des gesamten Projekts. Hiermit können Probleme schnell erkannt und gegebenenfalls auch vermieden werden. Ein weiteres wichtiges Diagramm stellt das Velocity Chart dar, das den abgearbeiteten Aufwand eines Sprints angibt und auswertet. Ob ein Tool über so ein Chart verfügt und wie verständlich die Erstellung ist, wird unter den Punkten „Release Burndown Chart", „Sprint Burndown Chart" und „Velocity Ermittlung & Chart" analysiert und bewertet.

Das „Sprint Planning" stellt eines der wichtigsten Kriterien dar. Wie einfach Sprints im Tool angelegt und anschließend auch bearbeitet und verwaltet werden können, wird hier bewertet. Ob das zu bewertende Tool noch weitere Reportings oder Zusatzfunktionen mit sich bringt und wie nützlich diese für die Unterstützung eines Teams sind, wird zusätzlich berücksichtigt.

3.1.2. Kanban Kriterien

Zur leichtgewichtigen Softwareentwicklung gehören zusätzlich zu Scrum- auch Kanban-Funktionen. Deshalb wurde eine Kriteriengruppe zum Punkt „Kanban" erstellt, anhand der die einzelnen wichtigen Punkte, die Kanban betreffen, bewertet werden.

Zuerst wird die Gestaltung des Kanban-Boards untersucht. Das Board ist das zentrale Element bei der Nutzung der Kanban-Methodik, da es den Workflow und das Work in Prozess (WIP) visualisiert. Bei der Bewertung geht es einmal um die Übersichtlichkeit sowie das Design des Boards. Es sollte auch mit einbezogen werden, wie viele Funktionen direkt auf dem Board verfügbar sind. Für Kanban von Bedeutung ist auch die Anzeige von Hierarchien der Prozesse, zum Beispiel welche Aufgaben zu welchem Überpunkt oder Projekt gehören. Wird die Schachte-

lung der Prozesse verständlich dargestellt, sollte hier eine hohe Punktzahl vergeben werden. Beim Cumulative Flow Diagram Kriterium sollte zuerst herausgefunden werden, ob das jeweilige Tool dieses Diagramm unterstützt. Wenn ja, sollte die Bewertung auf die Verständlichkeit und das Design des Diagramms abzielen.

Die WIP-Organisation wird bewertet, indem auf die Grenzen der parallel laufenden Arbeiten geachtete wird und ob diese sinnvoll gesetzt werden. Ein wichtiger Faktor des Kanban ist die Limitierung von Aufgaben in den einzelnen Spalten des Prozesses, um den Workflow zu verbessern.

Das Kriterium Kartengestaltung bezieht sich auf die einzelnen Karten auf dem Kanban-Board, welche Aufgaben darstellen. Hier ist die Frage, ob die Karten übersichtlich gestaltet sind und ob sie ermöglichen, ausreichend Informationen über jede Aufgabe anzugeben. Wie auch bei der Kategorie „Scrum" wird auch bezüglich der Kanban Funktionen berücksichtigt, ob das Tool weitere Reports oder Diagramme anbietet oder ob es zusätzliche nützliche Funktionen gibt.

3.1.3. Kollaboration

Die Kategorie „Kollaborationsfunktionen" bewertet die Möglichkeiten des Tools die Zusammenarbeit des Teams zu unterstützen.

Im Punkt „Online Community" wird betrachtet, ob eine Online Community bereits besteht und inwiefern diese entwickelt, ist um den Austausch von Informationen und Erfahrungen zum Tool zu ermöglichen.

Des Weiteren werden die teaminternen Kollaborationsfunktionen beurteilt. Dazu gehört der Unterpunkt „Nachrichtensystem", welcher die E-Mail bzw. Chat-Funktionen des Tools bewertet. Zudem wird geprüft, ob ein gemeinsam nutzbarer Team-Kalender zur Verfügung steht und ob eine Feedbackfunktion implementiert ist, um Kommentare zu den Tasks verfassen zu können.

3.1.4. Implementierung

Die Kategorie „Implementierung" befasst sich mit den Kosten und dem anfallenden Arbeitsaufwand für die Implementierung und Instandhaltung des Tools. Es werden der initiale Kaufpreis sowie mögliche laufende Kosten während der Nutzungszeit betrachtet. Der Punkt „Installation" bewertet die Komplexität der initialen Installation des Tools. Hier werden auch die Systemanforderungen der Hardware und Software des Tools an den User bewertet. Unter dem Punkt

„Hosting/Licensing Optionen“ wird betrachtet, ob die Applikation lokal gespeichert wird oder ob auf das Tool über eine Web-Anwendung zugegriffen werden kann. Der Unterpunkt „Instandhaltung“ bewertet den Aufwand der Instandhaltung des Tools durch beispielsweise. häufige Updates.

3.1.5. Plattform

Beim den Kriterien unter dem Überpunkt „Plattform“ handelt es sich um nicht-funktionale Aspekte des Tools, welche die Anwendung allgemein betreffen. Hier geht es darum, ob das Tool eine übergreifende oder beschränkte Suchfunktion besitzt und wie diese benutzbar ist. Außerdem wird die mobile Nutzung der Anwendung unter diesem Punkt bewertet. Gibt es eine Lösung zur Nutzung auf mobilen Geräten, und wenn ja, ist diese optimiert und immer verfügbar?

Unter dieser Kategorie sollen auch die Sicherheit und die Stabilität des Tools untersucht werden. Bei der Sicherheit soll in Betracht gezogen werden, ob der Herausgeber sich um besondere Sicherheit der Daten des Kunden kümmert, wie etwa wenn das Tool auch als private Cloud/Server Lösung angeboten wird. Die Stabilität betrifft die Nutzung, das Laden der Daten und eventuelle Abstürze.

Beim Kriterium „Verbreitungsgrad“ wird bewertet, wie weit das Tool auf dem Markt verbreitet ist. Dies steht auch im Zusammenhang mit der Größe der Nutzercommunity und Weiterentwicklung (neuen Releases) der Anwendung. Eine hohe Verbreitung gibt den Hinweis für gute Qualität.

Als Kriterien für die Unterstützung der User wurden „Support“ und „Trainings“ gewählt. Es wird bewertet, wie weit der Support ausgeprägt ist und ob dieser immer verfügbar ist. Dies ist ein wichtiger Punkt, falls Probleme bei der Nutzung auftreten. Bei der Bewertung von Trainings sollte eine hohe Bewertung vergeben werden, wenn es (viele) Trainings gibt und diese auf den Nutzer abgestimmt sind. Sie sollten helfen, die grundlegenden Funktionen des Tools zu verstehen, bevor mit der Nutzung begonnen wird.

3.1.6. Schnittstellen

Da viele Unternehmen bereits mit vorigen Projektmanagement-Tools ihre Projekte unterstützt haben, ist die Integration und Erweiterbarkeit mit neueren und anders spezialisierten Tools relevant. In der Kriteriengruppe „Schnittstellen“ wird unter dem Punkt „Erweiterbarkeit/Integration“

bewertet, ob ein Tool Funktionalitäten für die Kombination mit anderen Anwendungen und die Erweiterbarkeit durch diese anbietet.

Einige Tools für die leichtgewichtige Softwareentwicklung können durch zusätzliche Apps des Herstellers erweitert werden. Es kann jedoch passieren, dass die gewünschte Zusatzfunktion nicht durch bereits entwickelte Apps abgedeckt wird und aus diesem Grund eigene Ideen in Apps umgesetzt werden möchten. Ob und zu wie vielen Entwicklungsumgebungen die Entwickler eine Verbindung herstellen können, wird in dem Kategoriepunkt „Integration mit Entwicklungsumgebungen" berücksichtigt. Mit dem Punkt „Import/Export Funktion" wird bewertet, inwiefern Daten und Dokumente ins System importiert und auch aus dem System exportiert werden können.

3.1.7. Customization

Die Kriteriengruppe „Customization" beschäftigt sich mit Kriterien wie beispielsweise der „Benutzerverwaltung". Dieses Kriterium gibt Informationen darüber, ob es eine Begrenzung bei der Erstellung und Anmeldung von Benutzern für ein Projekt gibt und wie viele Angaben zu den einzelnen Benutzern verwaltet werden können. Der Punkt „Sprache" zeigt, ob das Tool in der gewünschten Sprache oder sogar in mehreren Sprachen erhältlich ist.

Ob sich das Tool intuitiv und einfach bedienen lässt oder ob vor der Nutzung Trainings notwendig sind, damit man das Projekt starten kann, wird mithilfe des Kriteriums „Intuitive Bedienung" analysiert. Inwiefern ein Tool auf die persönlichen Anforderungen und Vorstellungen angepasst werden kann, wird unter dem Punkt „Anpassbarkeit/Custom Fields" berücksichtigt.

3.2. Anwendung des Kriterienkatalogs

Die ausgewählten Kriterien können in einem Excel-Dokument bewertet werden. Das Excel-Dokument teilt sich in zwei Kriterienblöcke: funktionale und nicht-funktionale Anforderungen an das Tool. Jeder Kriterienblock setzt sich aus den dazugehörigen Unterkriterien zusammen (siehe oben). Da jedes Team bzw. an jedes Projekt unterschiedliche Anforderungen gestellt werden und daher die Gewichtung der Kriterienblöcke und einzelnen Unterkriterien individuell ist, be-

steht die Möglichkeit, die prozentuale Gewichtung im Dokument entsprechend anzupassen. Anschließend können zwischen einem und drei Tools gleichzeitig bewertet werden. Mithilfe der Gewichtung entsteht zum Abschluss eine Rangfolge der Tools. Dadurch soll es dem Benutzer ermöglicht werden, schnell das geeignete Tool für sein Projekt zu finden. Jedes einzelne Kriterium wird zunächst bewertet. Neben der Bewertung kann ebenfalls eine Begründung für die ausgewählte Bewertung eingetragen werden. Die Bewertung der einzelnen Kriterien erfolgt mit den Nummern 0-10. Die Zahl 0 wird ausgewählt, wenn eine Funktion im Tool überhaupt nicht vorhanden ist und die Zahl 10 ist das beste Ergebnis, das erreicht werden kann. Ein Kriterium mit der Bewertung 5 zeigt, dass dieses Kriterium im System erfüllt bzw. vorhanden ist. Sollte dieses nun vorhanden, aber eher schlecht ausfallen, wird eine Zahl zwischen 1 und 4 ausgewählt. Ist das Kriterium vorhanden und auch gut bewertet, liegt die Bewertung zwischen 6 und 9.

Nachdem das Excel-Sheet geöffnet wurde, kann man im Dokument rechts auf den Button „Start" drücken. Mithilfe dieses Buttons können die für die Bewertung erforderlichen Einstellungen getroffen werden. Zuerst werden dem Nutzer die Instruktionen für die Nutzung des Bewertungssheets angezeigt. Anschließend kann der Nutzer auswählen wie viele Tools er bewerten und vergleichen möchte. Es können maximal drei Tools verglichen werden. Nachdem die Anzahl der Tools ausgewählt wurde, können folgend die Namen der Tools eingegeben werden. Im nächsten Schritt der Einstellungen können die einzelnen Kriterien der übergeordneten Kriterienbereiche gewichtet werden. Die zwei Kriterienbereiche stellen funktionale und nicht-funktionale Anforderungen dar. Die Summe der prozentualen Anteile der einzelnen Kriterien muss in jedem Kriterienbereich am Ende in der Summe 100% ergeben. Nachdem alle Gewichtungen eingetragen wurden, wird mit einem Klick des Buttons „Fortfahren" die Tabelle im Excel-Sheet mit den eigegebenen Daten gefüllt. Die Namen der Tools und auch die passenden Gewichtungen werden an die richtige Stelle in der Tabelle eingetragen. Abbildung 7 zeigt den Aufbau des Worksheets.

Nun kann der Benutzer mit der Bewertung der Tools beginnen, um am Ende dasjenige zu finden, das für sein Projekt bzw. Team am besten geeignet ist.

Vergleich von Tools zur leichtgewichtigen Softwareentwicklung							
Kriterien		JIRA Agile			ALM Platform		
Funktionale Anforderungen	Gewichtung	Begründung	Bewertung	Score	Begründung	Bewertung	Score
Scrum	40,00%			9,333333			10,88889
Backlog-Erstellung	5	multi-select ra	6	30		5	25
Backlog-Items	5	User Story Man	5	25	User Story Man	5	25
Task Board	5	Swimlanes auf	6	30	iteration overv	6	30

Abb 7: Ausschnitt aus Bewertungstabelle

4 Vergleichsstudie der leichtgewichtigen Produkte

4.1. Kommerzielle Tools

4.1.1. HP – Agile Manager

Der Agile Manager von HP basiert auf der Software as a Service (SaaS) Technologie und dient als teamübergreifendes Planungswerkzeug. Der Agile Manger von HP unterstützt Scrum sowie Kanban. Der Fokus liegt jedoch auf Scrum.

In Bezug auf Kollaborationsfunktionen ist der Agile Manager noch ausbaufähig. Ein dediziertes Entwicklernetzwerk besteht beispielsweise nicht. In Sachen der Implementierung ist besonders die einfache Installation zu bemerken, da diese durch eine Web-Applikation entfällt und lediglich eine Anmeldung erfolgen muss. Eine einmonatige Testversion ist online verfügbar. Für die Vollversion fallen monatliche Nutzungsgebühren von 39 US$ an. Das Tool von HP weist gute Sicherheitsstandards auf und kann auch in Hinsicht auf Stabilität punkten. In Punkto Verbreitungsgrad lässt sich der Agile Manager zu den „Challengern" im agilen Umfeld zählen.[34]

Der Agile Manager basiert auf einer Web-Anwendung die vor allem durch ihre intuitive Bedienung punktet. Die Web-Applikation verfügt über eine einfache und individuell anpassbare Oberfläche. Die Anwendung arbeitet verstärkt mit visuellen Elementen wie Grafiken zur Unterstützung der Übersichtlichkeit (vgl. Abb. 8)

[34] Vgl. Duggan, J./Murphy, T. E. (2012), Abruf: 05.01.2015

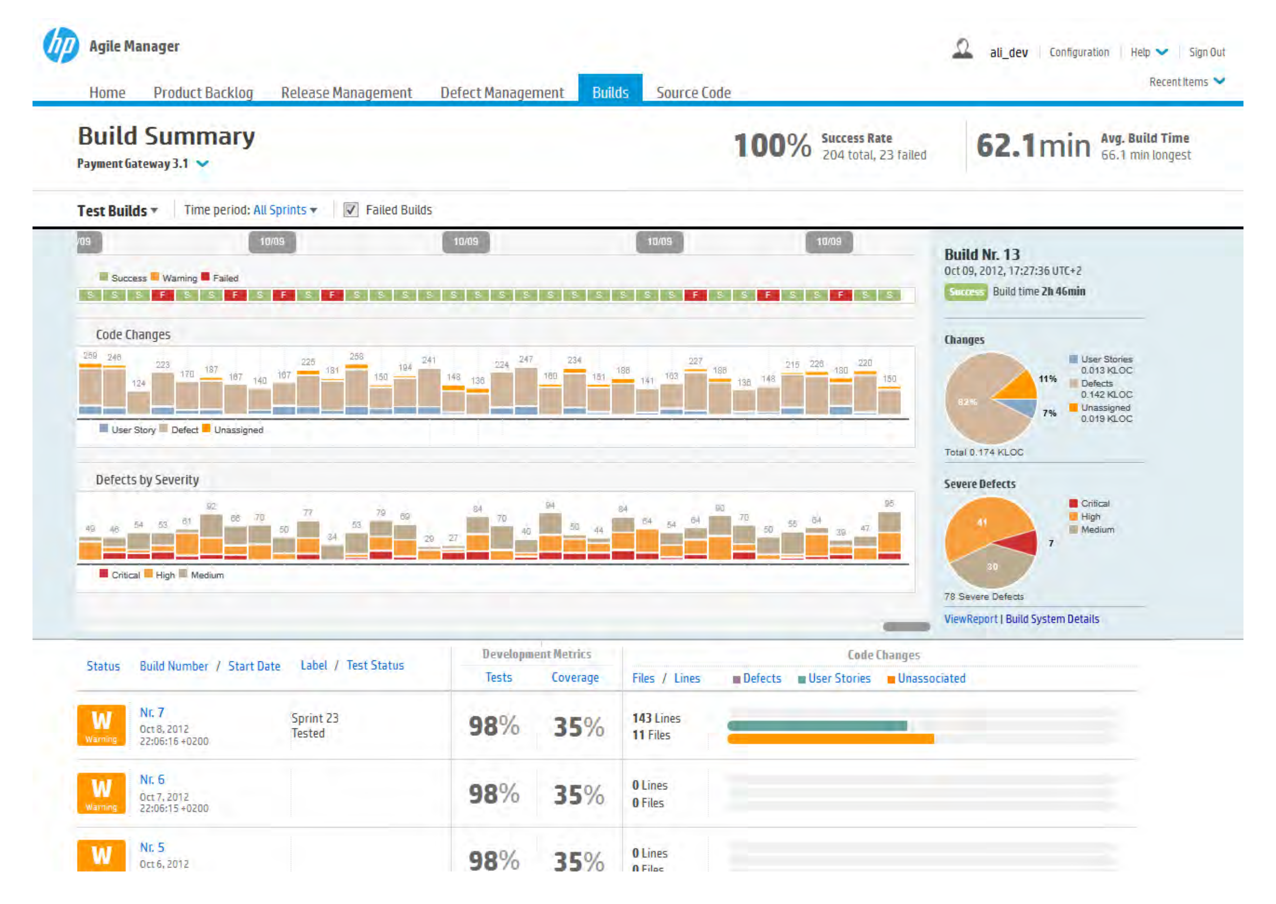


Abb 8: Build Summary[35]

4.1.2. IBM – Agile Development

„Agile Development" ist ein kommerzielles Tool der Firma IBM mit einem hohen Verbreitungsgrad auf dem Markt. Das Tool unterstützt sowohl Scrum als auch Kanban Ansätze.

Zu den implementierten Scrum-Funktionen gehören Produkt- und Release Backlogs, gut strukturierte Task Boards (vgl. Abb. 9) und Übersichts-Charts zum Release Burndown, Sprint Burndown und zur Velocity-Ermittlung. In Bezug auf Kanban-Eigenschaften verfügt das Tool über eine Kanban Boardgestaltung, ein Cumulative Flow Diagramm sowie WIP-Limits.

Besonders stark ausgeprägt sind die Kollaborationsfunktionen des Tools. Im Tool implementiert ist beispielsweise ein Team-Chat mit Anwesenheitsstatus der Teammitglieder. Kollaborationswerkzeuge wie Skype und Google Talk wurden in die Anwendung integriert.

[35] Enthalten in: Hewlett-Packard Development Company, L.P. (2015), Abruf: 02.12.2014

Die Preise für das IBM-Tool variieren stark in Abhängigkeit der Nutzerzahl. Eine einzelne Installation kostet 214 Euro, während für Unternehmensweite Installationen mit Kosten bis zu 2900 Euro gerechnet werden müssen. Die Installation der Anwendung ist zum Teil sehr komplex und zeitaufwendig.

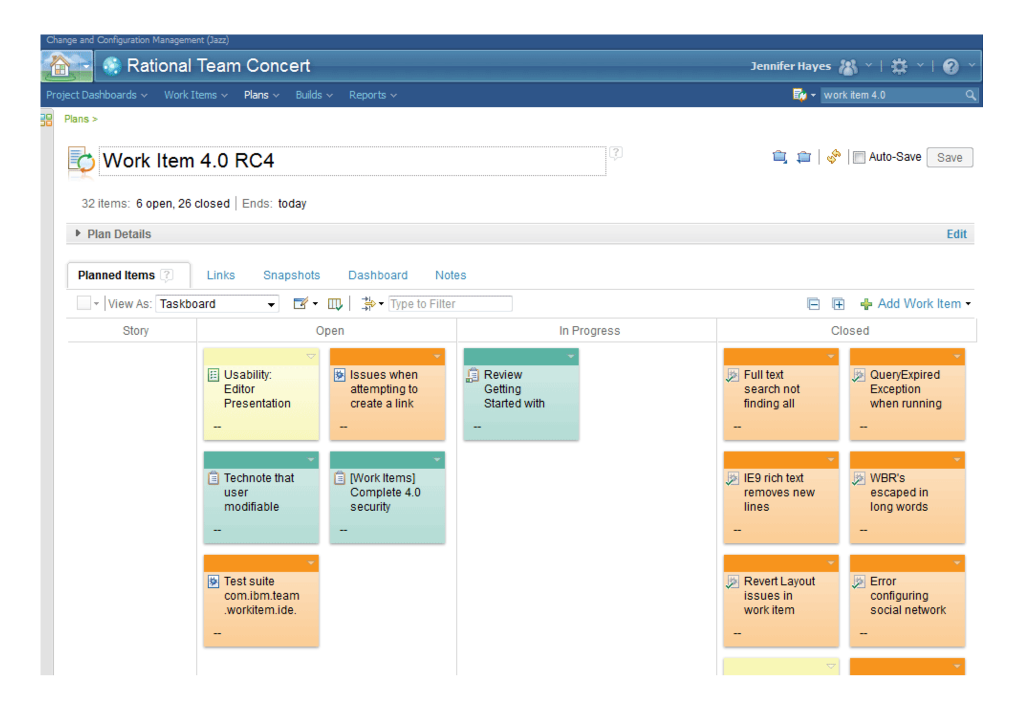


Abb 9: Taskboard[36]

4.1.3. Digité – SwiftKanban

Bei SwiftKanban von Digité handelt es sich um ein kommerzielles, webbasiertes Tool, das über den Browser aufgerufen wird. Die Anwendung ist besonders für die Anwendung der Kanban-Methodik optimiert.

Die Arbeit kann in verschiedene Projekte eingeteilt werden, wobei jedes Projekt ein separates Team, Backlog und Kanban-Board haben kann. Es gibt ein Kanban-Board für jedes Projekt, auf dem die Karten in Spalten organisiert sind. Hier ist ein Standard-Prozess definiert, welcher angepasst werden kann. Neue Spalten und Lanes können angelegt und das WIP-Limit bestimmt werden. Karten können direkt neu angelegt oder aus dem Backlog hinzugefügt werden. Beim Klick auf eine Karte können Aktionen über ein schnelles Menü ausgeführt werden (vgl. Abb. 10). Außerdem kann die Karte „umgedreht" werden, sodass ein Bearbeitungsverlauf angezeigt wird.

[36] Enthalten in: IBM (2014), Abruf: 01.12.2014

Eine Besonderheit bei SwiftKanban ist, dass Karten langsam ausgeblendet bzw. unsichtbar werden, wenn sie länger nicht aktualisiert wurden.

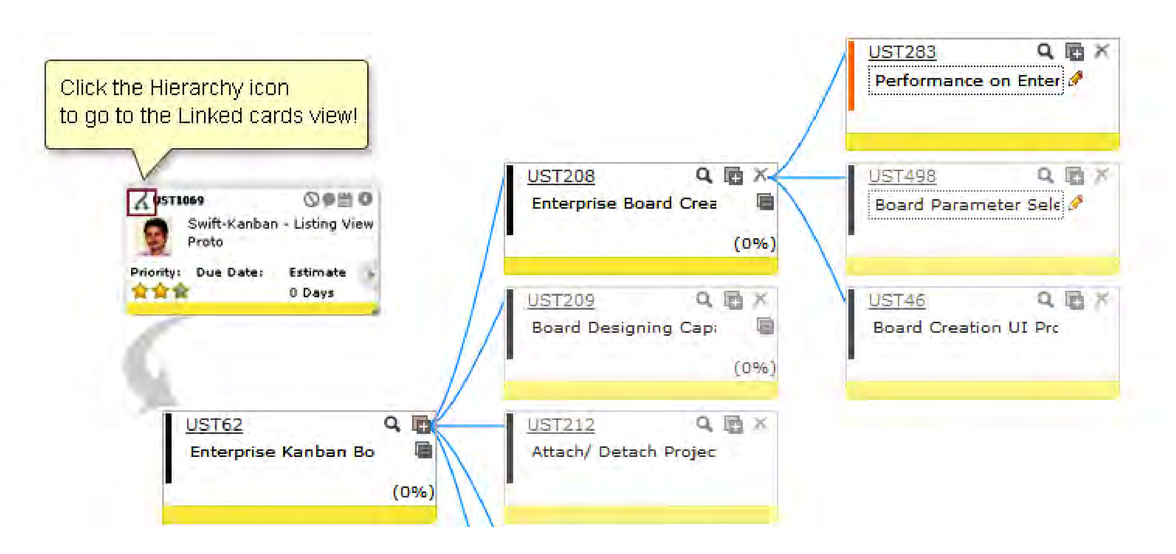


Abb 10: Linked Cards Ansicht in SwiftKanban[37]

Der Board-Filter ermöglicht es, das Board aus der Sicht verschiedener User zu betrachten und deren Workload anzuzeigen. Es wird außerdem ein personalisierbares Dashboard mit Widgets für jeden User angeboten. Für die Kollaboration im Team bietet Swiftkanban die Möglichkeit für Diskussions-Threads sowie eine direkte Chatfunktion. Den Karten können auch Kommentare hinzugefügt werden, um andere Teammitglieder auf Besonderheiten aufmerksam zu machen. Tritt ein Problem mit einer Karte auf, kann diese geblockt werden, sodass das gesamte Team das Hindernis erkennen kann. Verschiedene Reports und Diagramme zum Cumulative Flow, Lead Time und Cycle Time können aufgerufen werden. Für Teams, die ebenfalls Scrum nutzen, bietet das Tool außerdem ein Scrum Release Planning sowie eine User Story Hierarchie/Breakdown Funktion. Ebenso gibt es Burndown-Charts.

Für SwiftKanban Nutzer gibt es die Versionen Free (bis 20 User), Team (7€) und Enterprise (15€). Vor dem endgültigen Kauf kann das Produkt für 30 Tage getestet werden. Die Registrierung kann entweder über das Anlegen eines neuen Benutzerkontos erfolgen, aber auch ein Login über Facebook oder Google+ ist möglich, was den Vorgang beschleunigt.

Zusammenfassend bietet SwiftKanban alle Basis-Funktionen für die Anwendung des Kanban an. Das Tool ist übersichtlich und einfach aufgebaut und die angebotenen Diagramme helfen, den Fortschritt nachzuverfolgen. Besonders die vorhandene Chatfunktion hilft dem Team bei der

[37] Enthalten in: SwiftKanban Inc. (2015), Abruf: 01.12.2014

Zusammenarbeit. Hilfreich bei SwiftKanban ist die zusätzliche Einbindung einzelner Scrum-Aspekte, welche es für das leichtgewichtige Scrumban optimieren. Allerdings fehlt eine angepasste App zur mobilen Nutzung, welche in Zeiten der durchgängigen Smartphone und Tablet Nutzung sehr wertvoll wäre.

4.1.4. Leankit – Leankit

Bei Leankit von der gleichnamigen Firma handelt es sich wie bei Swiftkanban um ein kommerzielles online Tool zur Unterstützung der Kanban-Methode. Das Leankit-Team erklärt, dass das Tool durch einen am Menschen orientierten Ansatz der Produktivität die Art, wie Teams ihre Arbeit erledigen, revolutionieren wird.[38]

Das Tool bietet ein sehr übersichtliches Kanban-Board mit anpassbaren Karten. Die einzelnen Karten können bezüglich des Verantwortlichen, der Wichtigkeit, möglicher Probleme und des Status spezifiziert werden. Verschiedene Farben der Karten helfen zusätzlich, diese zu kategorisieren. Abbildung 11 zeigt die Details, die zu jeder Karte angegeben werden können.

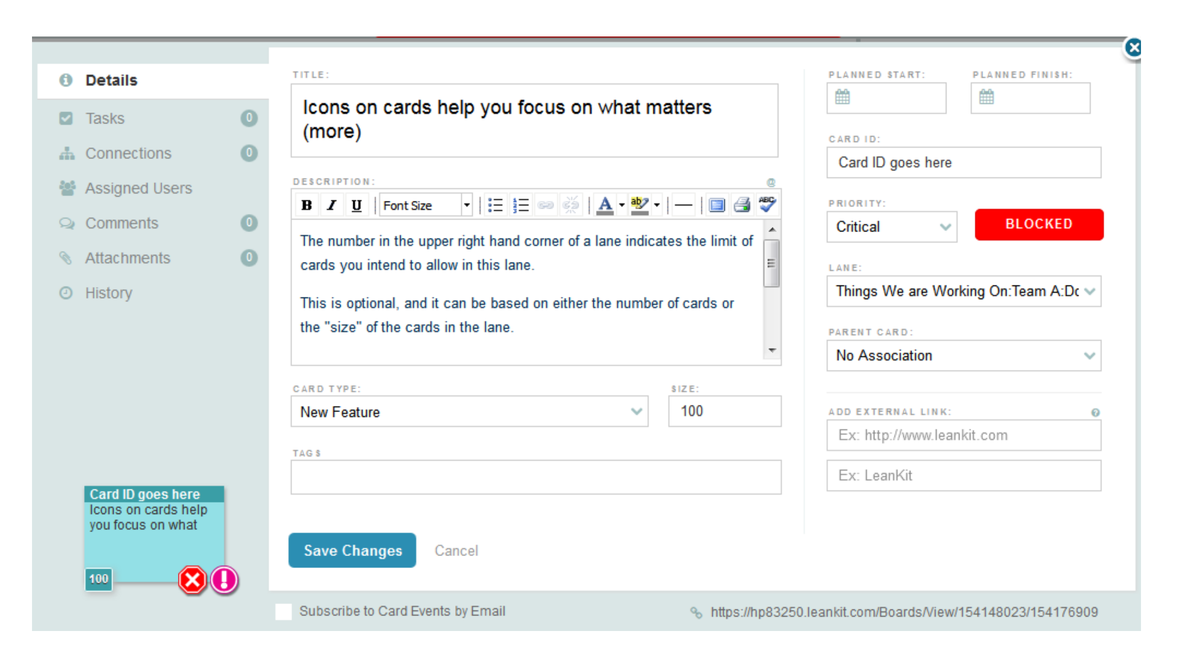


Abb 11: Details einer Karte[39]

Die Karten werden dann, wie beim Kanban üblich, anhand des Status auf dem Board in Spalten angeordnet. Außerdem werden in den Spalten horizontale „Swim Lanes“ angezeigt, um parallele

[38] Vgl. LeanKit Inc. (2014a), Abruf: 02.12.2014
[39] Enthalten in: LeanKit Inc. (2014a), Abruf: 02.12.2014

Abläufe zu verdeutlichen. Das Drag&Drop-System und die mögliche Minimierung der Spalten machen die Bedienung sehr intuitiv, wobei auch jeder Schritt eines Users in der „History" nachverfolgt werden kann. Der sogenannte „Activity Stream" kann zusätzlich einfach über die Menüleiste aufgerufen werden und zeigt alle Aktionen an.

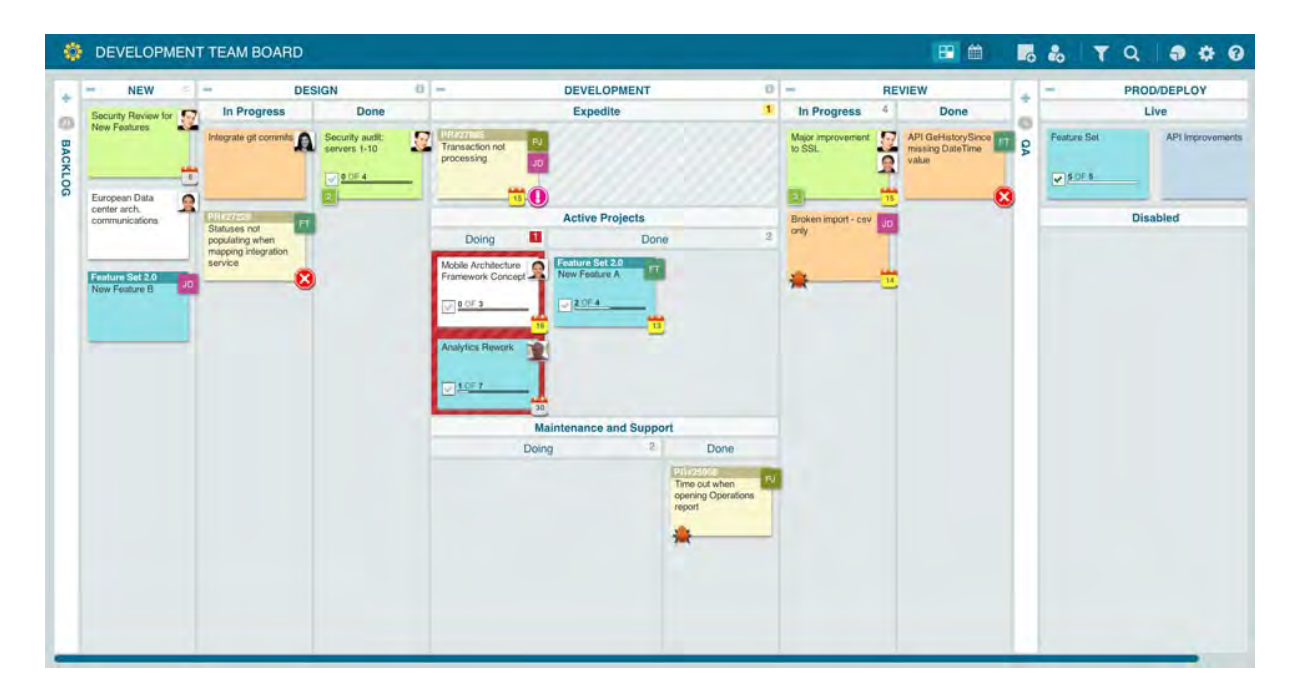


Abb 12: Leankit Kanban Board[40]

In den Einstellungen kann das Layout des Kanban-Boards (vgl. Abb. 12) angepasst werden. Außerdem sind weitere Einstellungen bezüglich der Zugriffsrechte und der verfügbaren Kartentypen möglich. Hier gibt es auch die Möglichkeit, Karten (als CSV-Dateien) zu im- oder exportieren.

Es gibt für das Zeitmanagement eine Kalenderübersicht und Fälligkeitstermine für die Aufgaben eines jeden Users. Hier kann vom Hauptboard durch einen Klick einfach zur Kalender-Ansicht gewechselt werden. Außerdem kann Leankit einfach von mobilen Geräten aus genutzt werden, wobei Updates sofort für alle Team-Mitglieder sichtbar sind. Die optimierte Leankit App vereinfacht die Nutzung auf Smartphones und Tablets, sodass schnell von überall auf laufende Projekte zugegriffen werden kann. Bezüglich der Kollaboration bietet Leankit die Möglichkeit zu einzelnen Elementen Kommentare abzugeben. Diese können dann als Feedback für das Team genutzt werden. Zusätzlich können im Rahmen von Abonnements Benachrichtigungen per Mail

[40] Enthalten in: LeanKit Inc. (2014a), Abruf: 02.12.2014

eingerichtet werden, um bezüglich einer bestimmtem Aufgabe oder eines Projektbereichs direkt informiert zu werden. Ein direktes Nachrichtensystem zwischen Nutzern ist nicht vorhanden.

Das Tool bietet auch verschiedene Reports und Diagramme an, die es dem User erleichtern, den Fortschritt zu überblicken. Es gibt Diagramme zum Cumulative Flow, Kartenverteilung, zur Durchlaufzeit, Effizienz und zur Prozesskontrolle.[41] Es gibt eine Suchfunktion, bei der die Suche durch verschiedene Filter wie Datum, Bereich oder User eingegrenzt werden kann.

Leankit bietet ein flexibles Integrationsframework, welches die Verbindung zu anderen Anwendungen ermöglicht. Zu diesen zählen etwa JIRA, MS Visual Studio/TFS und MS Visual Studio Online, MS Project Server, GitHub und Oracle Primavera. Außerdem können Verbindungen mit über 250 Web-Apps, wie zum Beispiel Google, Salesforce und Zendesk, aufgebaut werden (dazu nutzt das Tool Zapier).

Auf der Website wird eine detaillierte Produkt-Demo angeboten, die einen ersten Überblick über die Funktionen gibt. Weiterhin ist eine User-Community und online Knowledge-Base mit Support-Angebot vorhanden. Auch gibt es die Möglichkeit vor der Kaufentscheidung eine Testversion für 30 Tage auszuprobieren. Leankit gibt es in den Versionen Basic (kostenlos), Team (15€/User) und Portfolio (19€/User).

Alles in allem beinhaltet Leankit alle nötigen Kanban-Funktionen, die ein Anwender-Team benötigt. Verglichen mit SwiftKanban ist das Kanban-Board ausgereifter und durch die Farben und Kartendetails übersichtlicher. Es ähnelt der Nutzung von Post-Its auf einem Board und ist damit sehr intuitiv. Besonders sticht bei Leankit auch heraus, dass es mit zahlreichen anderen Anwendungen kombiniert werden kann. Dieses Tool bietet für kleine und große Teams, welche Kanban in den Mittelpunkt ihrer Arbeit stellen, eine nützliche Unterstützung bei der Entwicklung.

4.1.5. Atlassian – JIRA Agile

Bei dem Softwareprodukt „JIRA Agile", hergestellt von der Firma Atlassian, handelt es sich um ein Produkt, dass die wichtigsten und bewährtesten Merkmale von agilen Methoden vereinigt. JIRA Agile ermöglicht es, mehrere Projekte auf einem einzigen Board oder auch verschiedenen darzustellen, um einen guten Überblick über das Unternehmen und die einzelnen Projekte zu geben. Verschiedene „Swimlanes" auf den Boards geben dem Benutzer die Möglichkeit, den eigenen Fortschritt und den des gesamten Teams auf einen Blick zu sehen (vgl. Abb. 13). „JIRA Agile" bringt spezielle Filter für die Boards mit sich. Diese ermöglichen es Tasks oder User Sto-

[41] Vgl. LeanKit Inc.(2014b), Abruf: 10.12.2014

ries besonders hervorzuheben, die schnellstmöglich bearbeitet werden sollten. Das Programm lässt sich gut an die Wünsche und Anforderungen der einzelnen Benutzer anpassen, z.B. können auch eigene Filter erstellt werden. Außerdem können eigene Add-Ons angefertigt werden. Zusätzlich dazu werden von Atlassian bereits über 100 Add-Ons bereitgestellt, damit das Programm weiter individualisiert werden kann.[42] Das Programm unterstützt neben Scrum Eigenschaften, wie z.B. Backlog Management, Release- und Sprint Burndown Charts auch Kanban Eigenschaften. Es bietet zum Beispiel ein Kanban Board und auch Cumulative Flow Diagram an.

„JIRA Agile“ basiert auf dem früher erschienenen „JIRA“. Beide Produkte sind in 14 verschiedenen Sprachen erhältlich. Die Produkte zeichnen sich durch einen hohen Grad der Integration aus, d.h. dass sich der Benutzer Unterstützung von vielen Drittanbietern holen kann, wie z.B. Tasktop, Google Apps, salesforce.com, aber auch die Integration mit anderen agilen Produkten wird unterstützt. Jedoch werden bei „JIRA Agile“ komplexe Prozess nicht sehr gut unterstützt und es mangelt an einem einzigen Kontrollpunkt für Workflows. Außerdem gibt es kein allein integriertes Dashboard.[43] Das Produkt ist entweder als Download auf den eigenen Server erhältlich oder in einer Cloud verfügbar. Atlassian bietet einen sehr guten Support für das Produkt, aber ist nicht so stark in der Bereitstellung und Durchführung von Trainings.

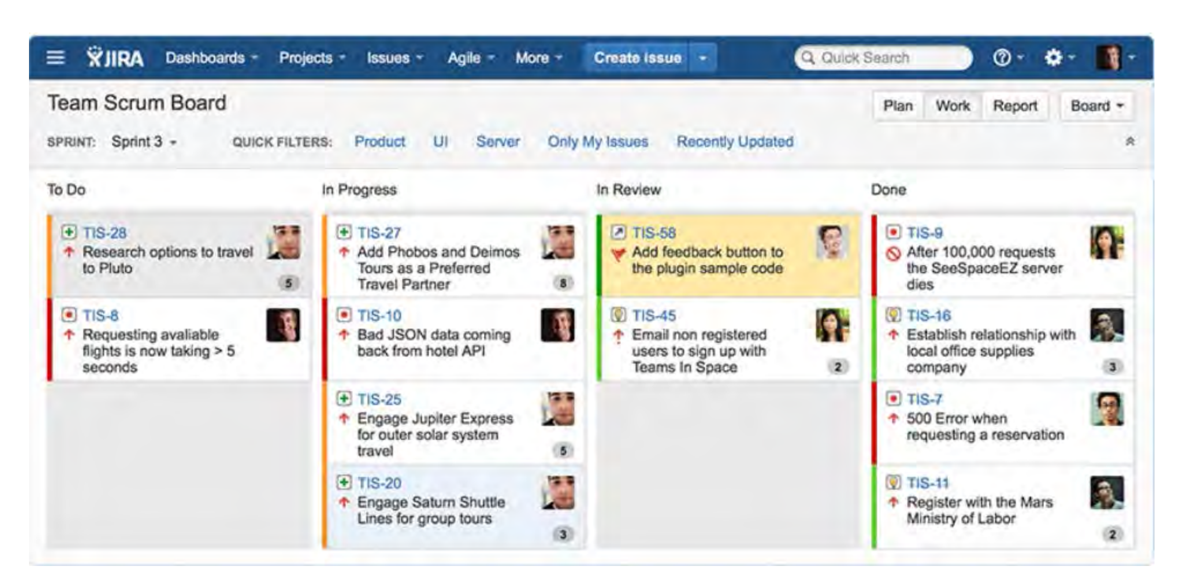


Abb 13: JIRA Scrum Board[44]

[42] Vgl. Atlassian Inc. (2015c), Abruf: 01.12.2014
[43] Vgl. Duggan, J./Murphy, T. E. (2012), Abruf: 05.01.2015
[44] Enthalten in: Atlassian Inc. (2015c), Abruf: 01.12.2014

4.1.6. Rally – ALM Plattform

Das Unternehmen Rally Software bietet das Produkt „ALM Platform“ auf dem Markt an. Das Programm unterstützt ebenfalls Kanban und auch Scrum Funktionalitäten. Zu den wichtigsten unterstützten Kanban Eigenschaften gehören zum Beispiel die Kanban-Karten. Diese ermöglichen es, Probleme und Verzögerungen festzustellen.[45] Damit die Qualität der eigenen Entwicklung und Aktivitäten festgestellt werden kann, liefert das Programm dafür „Defect-Charts“ und unterstützt auch das dazugehörige Reporting. Außerdem liefert die Plattform „Dashboards“, die den Fortschritt des gesamten Teams darstellen. Daraus lässt sich schließen, dass hier ebenfalls mehrere Teams und Projekte unterstützt werden. Außerdem können die Scrum und Kanban Teams ihren gemeinsamen Fortschritt auf einem gemeinsamen Board oder auch getrennt verfolgen. Damit das Team erkennen kann, wie lange eine Karte auf den Boards verharrt bzw. welche Karten vernachlässigt wurden, hat Rally das sogenannte „In-Column aging“ entwickelt.[46] Das Programm unterstützt z.B. User Stories, Sprint Planung und Management und bietet Release- und Sprint-Burndown Charts an.

Die ALM Platform zeichnet sich durch anpassbare Dashboards aus, die den Status des Projekts und den Arbeitsfluss sehr gut visualisieren, z.B. mit Hilfe von Burdown oder Velocity Charts (vgl. Abb. 14). Die Dashboards können mit dem Team geteilt werden, damit jeder auf demselben Stand ist. Dementsprechend zeichnet sich die Platform durch eine hohe Skalierbarkeit (Scalability) aus.

Auf der Rally-Homepage sind über 100 verschiedene Apps verfügbar, mit denen der Nutzer die Standardversion erweitern kann. Sollte die gewünschte App nicht dabei sein, können eigene Apps für die Plattform programmiert werden mit AppSDK.[47] Es ist zu beachten, dass Rally drei Versionen des Programms entwickelt hat, die unterschiedliche Anforderungen erfüllen: Rally Unlimited Edition, Rally Enterprise Edition und Rally Community Edition. Alle Versionen ermöglichen eine hohe Integrationsfähigkeit mit vorgefertigten Bindungen an die meist genutzten Tools, wie z.B. HP, Oracle und Jira). Sollte man sein eigenes System verwenden, liefert Rally hierfür ein hilfreiches Integrations Framework.[48] Das Produkt ist nur in Englisch verfügbar und nur als SaaS-Lösung erhältlich. Rally bietet dem Nutzer ein breites Spektrum an Trainings und Beratungsleistungen.

[45] Vgl. heise Developer (2010), Abruf: 05.12.2014
[46] Vgl. Rally Software Development Corp. (2015d), Abruf: 01.12.2014
[47] Vgl. Rally Software Development Corp. (2015b), Abruf: 01.12.2014
[48] Vgl. Rally Software Development Corp. (2015c), Abruf: 01.12.2014

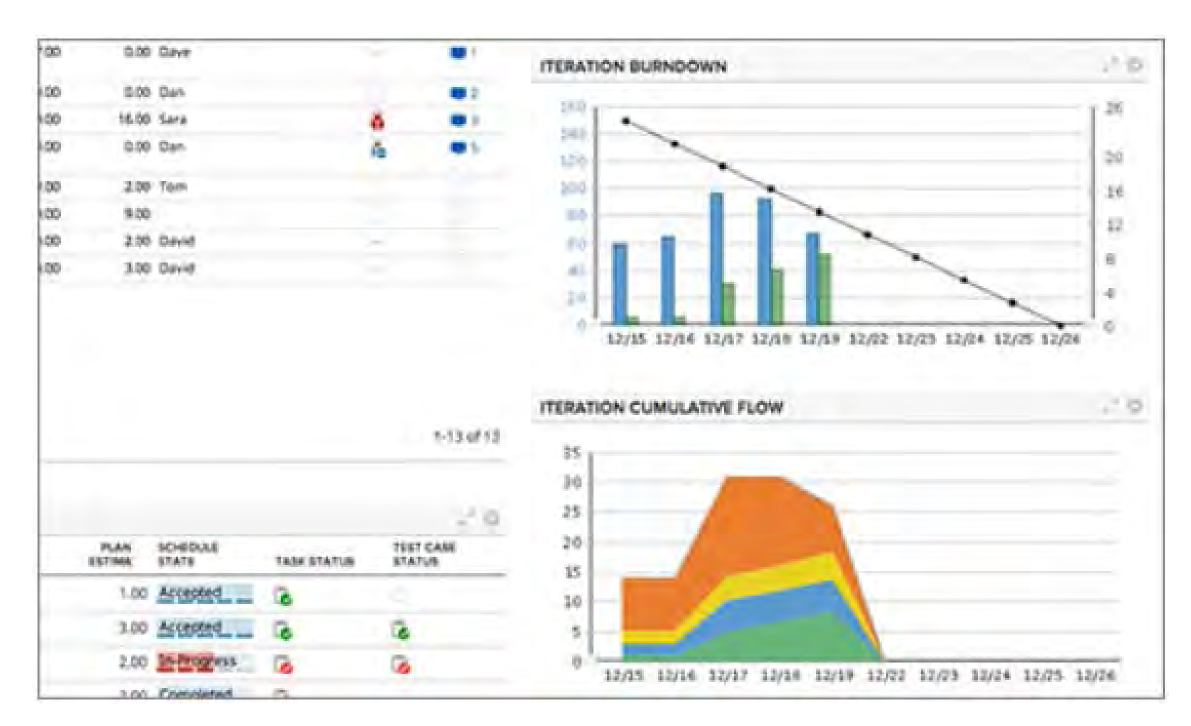


Abb 14: Rally ALM Platform[49]

4.2. Open Source Tools

4.2.1. KADOS

KADOS ist ein Tool, basierend auf einer Web-Applikation, das vorwiegend zum organisieren von Scrum-Projekten entwickelt wurde. Neben Scrum werden jedoch auch Kanban Methoden unterstützt. KADOS gehört zu den Open-Source Tools auf dem Markt und ist daher kostenlos erhältlich. Die gesamte Lösung von KADOS basiert auf der Verwendung von Post-Its.

Scrum Projekte werden in KADOS in drei verschiedene Level unterteilt: Projects, Releases und Sprints. Dabei verfügt jedes Level über ein eigenes Backlog. Kanban wird in KADOS durch die Anordnung von Post-Its auf mehreren Dashboards implementiert (vgl. Abb. 15).

KADOS muss vom Benutzer auf dem eigenen Server installiert werden und benötigt zur Implementierung eine MySQL Datenbank sowie einen Webserver. Das Tool verfolgt durch die Post-It-Struktur einen sehr visuellen Ansatz, unter welchem jedoch die Übersichtlichkeit leidet.

[49] Enthalten in: Rally Software Development Corp. (2015a), Abruf: 01.12.2014

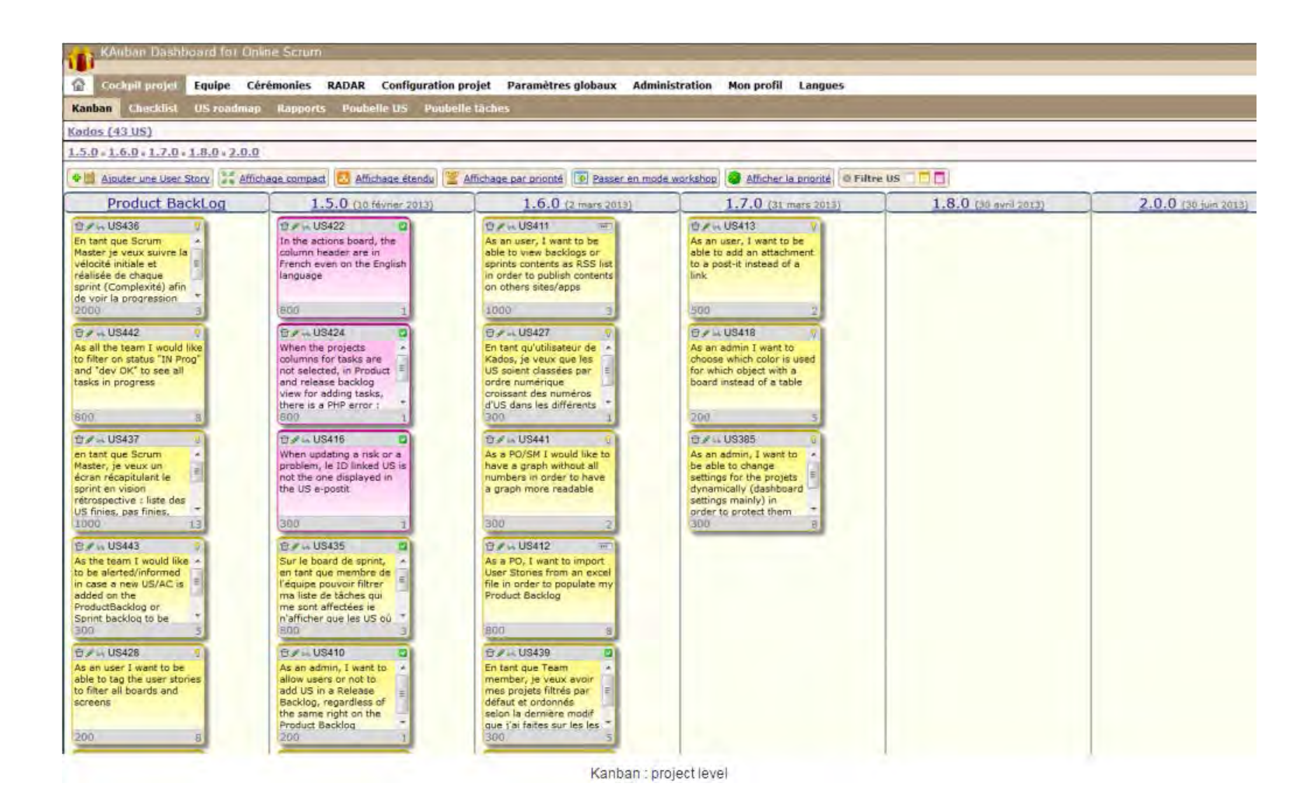


Abb 15: Project Level Overview[50]

4.2.2. AgileFant

Bei AgileFant handelt es sich um ein Open Source Tool zur leichtgewichtigen, genauer gesagt agilen, Softwareentwicklung. Es gehört zu den am weitesten verbreiteten agilen OS-Tools auf dem Markt. AgileFant unterstützt das „Scaled Agile Framework“, welches auf der folgenden Abbildung (Abb. 16) dargestellt ist, und bietet Features zur agilen Verwaltung von Portfolios, Projekten und Teams.[51]

[50] Enthalten in: Marmotte Technologies (o. J.), Abruf: 08.01.2015
[51] Vgl, Agilefant Ltd. (2015b), Abruf: 07.01.2015

Abb 16: Scaled Agile Framework[52]

Für bis zu fünf User kann die basis Version („Cloud“) kostenlos genutzt werden. Bei mehr als fünf Nutzern müssen hier $5 pro User im Monat gezahlt werden. Bei der Enterprise Version, die auf dem eigenen Server gehostet werden kann, sind es $20 pro Monat. Zusätzlich ist die Open Source Version auf GitHub verfügbar, welche allerdings einige Funktionen der bezahlten Versionen nicht beinhaltet. Diese kann jedoch, wie beim Open Source üblich, von den Entwicklerteams an die eignen Anforderungen angepasst werden.

Stories und Tasks werden in AgileFant notiert und die anfängliche „Original Estimate“ (OE) und spätere „Effort Left“ (EL) Zeiten werden notiert. OE ist die Zeit, die das Team am Anfang kalkuliert und die angibt, wie viel Zeit für eine spezifische Aufgabe benötigt wird. EL steht für die genauere Angabe, wie viel Zeit das verantwortliche Teammitglied noch für die Aufgabe braucht.

[52] Enthalten in: Leffingwell u. a. (2015), Abruf: 07.01.2015

Das Iteration Burndown Chart stellt grafisch dar, wie weit das Team fortgeschritten ist.[53] Das Tool bietet somit Unterstützung für die Anwendung der grundlegenden Scrum Prinzipien.

Das Backlog in AgileFant ist in drei Ebenen unterteilt: Produkt-Backlog, Projekt-Backlog und Iterations-Backlog (vgl. Abb. 17). Das Product Backlog in AgileFant besteht aus einem Baum („Tree"), welcher die Anforderungen des Produkts enthält. Der Story Tree eines jeden Produktes oder Projekts bietet eine hierarchische Übersicht über die verschiedenen Ebenen. Die Abbildung verdeutlicht die Hierarchie-Ebenen in AgileFant.

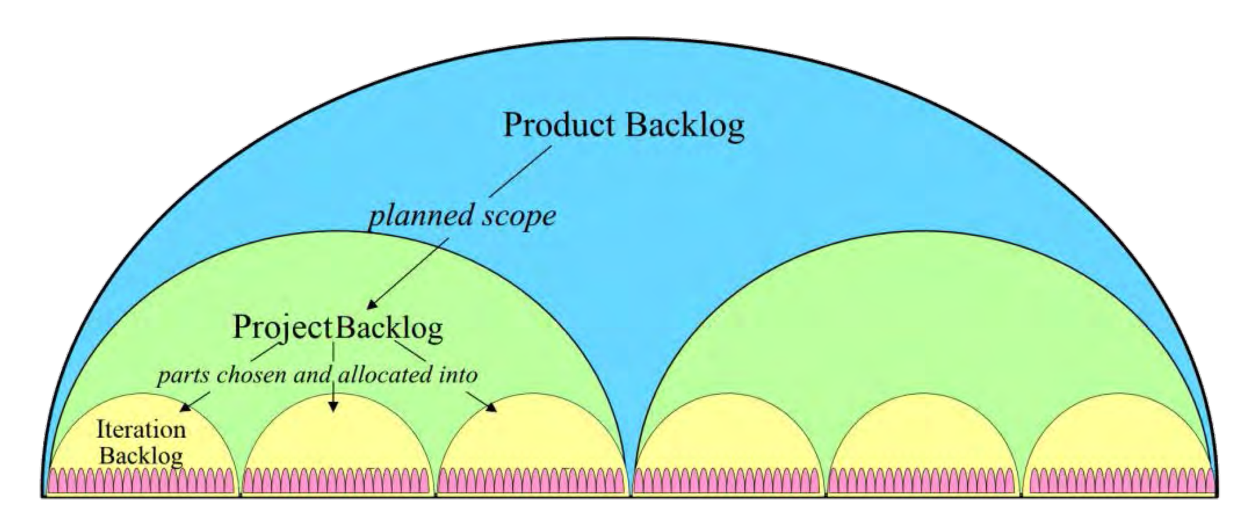


Abb 17: Backlog-Ebenen in AgileFant[54]

AgileFant bietet außerdem ein Zeiterfassungs-Feature (Timesheets) an, bei dem Stunden festgehalten werden können. Es gibt eine „Daily Work" Übersicht, die die aktuellen Aufgaben eines Users anzeigt. Dazu kommt das Portfolio View, welches alle aktuellen und geplanten Projekte und deren Status auflistet. Die erweiterte, nicht kostenfreie Version von AgileFant bietet eine App für mobile Geräte, sodass die Funktionen überall verfügbar sind.

Als Tool zur Unterstützung von Scrum bietet AgileFant alle notwendigen Funktionen. Die Oberfläche des Tools ist simpel gehalten und alle Funktionen sind direkt über die Startseite erreichbar. Die Anwendung von Kanban wird hier allerdings nicht unterstützt. Was heraussticht ist die Verbindung der Open Source mit der Entwicklungsumgebung GitHub, welche auch Zugang zu einer großen User-Community ermöglicht. Für die Nutzung im Unternehmen muss abgewägt werden, ob die kostenpflichtige Version mit mehr Funktionen einen größeren Nutzen bietet.

[53]Vgl. Jokelainen, R./Pekkanen, P. (o. J.), Abruf: 07.01.2015
[54] Enthalten in: Agilefant Ltd. (2015a), Abruf: 08.01.2015

4.2.3. IceScrum

Die Firma Kagilum bietet das Produkt „IceScrum” als Open Source Lösung für agile Softwareentwicklung an. Die Lösung kann für 14 Tage kostenlos als Cloud-Lösung getestet werden. Anschließend ist IceScrum zum einen als Cloud Lösung (IceScrum Cloud) und als Download (IceScrum Pro) erhältlich. Es handelt sich hierbei jedoch nicht um Freeware. Das Produkt ist eher für kleinere Teams als für große geeignet. Das Zusammenarbeiten der Teammitglieder wird allerdings sehr gut durch einen integrierten Chat und Feedback-Möglichkeiten unterstützt.

IceScrum bietet nicht nur Scrum Funktionalitäten, sondern bringt auch Kanban Funktionen mit sich, wie z.B. das Cumulative Flow Diagram und WIP-Organisation. Abbildung 18 zeigt einen beispielhaften, nach Kanban organisierten, Release Plan im Tool. Neben den gewöhnlichen User Stories können hier auch noch Technical Stories, für die technischen Entwicklungsaufgaben einer Software, und Story Defects, für aufgetretene Fehler bei der Entwicklung, erstellt werden. IceScrum kann mit den bekanntesten Source Code Management Tools integriert werden (SVN, Git). Zusätzlich ist die Integration zu den populärsten webbasierten Hosting Services wie GitHub und Google Code möglich. Das Tool bietet Exportfunktionen an, aber der Import eines vorigen Projekts auf den Kagilum Server ist nicht möglich.[55]

Das Unternehmen wächst in seiner Community und entwickelt IceScrum ständig weiter. Es gehört mittlerweile zu den wichtigsten Open Source Agile Tools. Vor ein paar Jahren wurden noch das Handling und die Bedienung des Tools stark kritisiert. Heute zeichnet es sich durch eine intuitive Bedienbarkeit aus. Nach Verfolgung von Kundenrezensionen auf Social-Media-Plattformen ergibt sich, dass es leider häufiger zu Problemen mit dem Datacenter und Netzwerkausfällen kommt.[56]

[55] Vgl. Kagilum SAS (2012), Abruf: 07.01.2015
[56] Vgl. Twitter (2014), Abruf: 08.01.2015

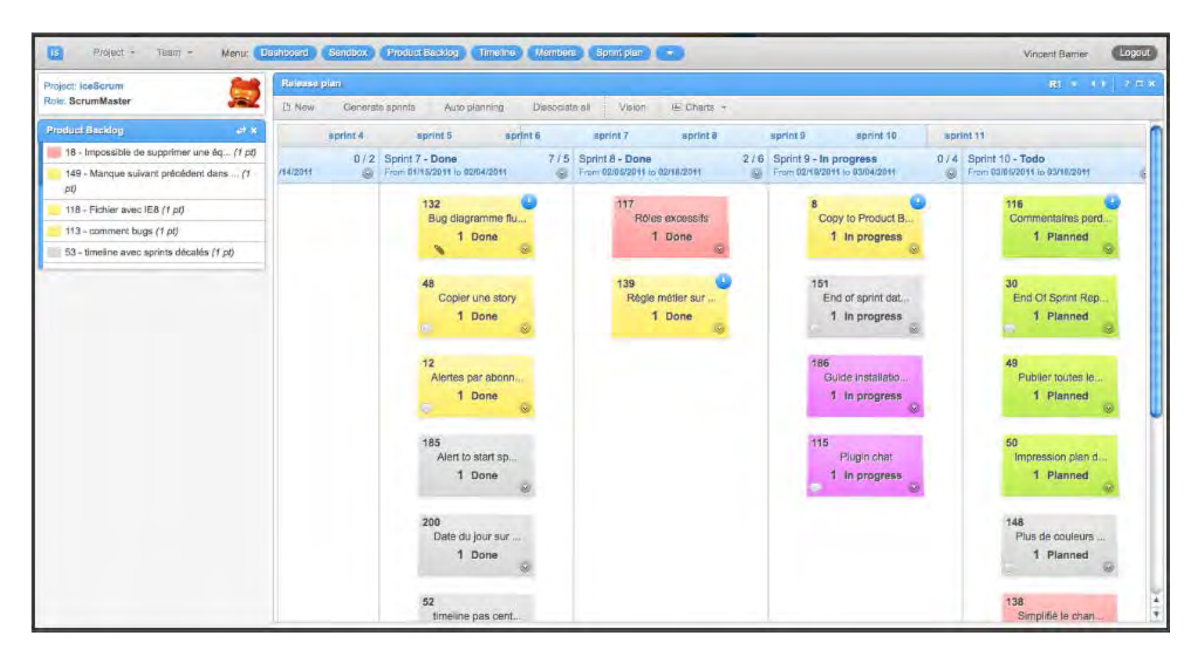


Abb 18: Release Plan - IceScrum[57]

4.3. Vergleich der Tools

Im folgenden Kapitel werden die bereits erläuterten Tools (vgl. Kapitel 4) miteinander verglichen. Dabei werden die Ausprägungen der oben beschriebenen Eigenschaften visualisiert und gegenübergestellt. Die Tabelle verdeutlicht anhand von farblicher Hervorhebung die positiven und negativen Eigenschaften der unterschiedlichen Tools. Grün steht für eine gute bis sehr gute Benotung, gelb zeigt eine mittelmäßige Benotung an und eine rote Färbung hebt negative Kritik an einem Punkt hervor.

Zusätzlich werden die wesentlichen Merkmale der Anwendungen hervorgehoben, um eine adäquate Entscheidung für ein Tool zur Projektunterstützung treffen zu können. Die Übersicht ermöglicht ein Verständnis der Hauptmerkmale auf einen Blick.

[57] Enthalten in: Kagilum SAS (2012), Abruf: 07.01.2015

Tool	Scrum	Kan-ban	Kol-labora-tion	Imple-ple-men-tierung	Platt-form	Schnitt stellen	Cus-tomi-zation
HP							
IBM							
Swift-Kanban							
Leankit							
Jira agile							
Rally							
KADOS							
AgileFant							
iceScrum							

Table 1: Vergleich leichtgewichtiger Tools

HP: Der Agile Manager von HP sticht durch seine einfache Implementierung und seinen hohen Grad an Customization hervor. Für Kanban ist das Tools jedoch nicht geeignet und auch die Kollaborationsfunktionen des Tools überzeugen nicht.

IBM: Die Vorzüge des IBM-Tools liegen im Scrum-Bereich und im Bereich der Team-Kollaboration. Nicht zu empfehlen ist die Applikation für reine Kanban-Projekte.

SwiftKanban: SwiftKanban bietet eine intuitive Bedienung und gute Kanban-Funktionen. Auch die Kollaboration ist durch Chat- und Kalender-Funktionen gut fundiert. Allerdings gibt es wenige Verbindungsmöglichkeiten mit anderen Tools und keine private Hosting Option.

Leankit: Leankit bietet eine gezielte Unterstützung der Kanban-Methodik durch ein übersichtliches, intuitives Kanban-Board und viele Reports. Scrum wird von diesem Tool nur sehr schwach unterstützt. Dafür bietet es eine eigene App, viele Integrationsmöglichkeiten und eine hohe Anpassbarkeit.

JIRA agile: Das Tool JIRA agile zeichnet sich durch seine hohe Anpassbarkeit an Kundenanforderungen aus. Jedoch Import- und Exportfunktionen sind nur sehr schwach unterstützt. Die Integration zu anderen Tools ist im Gegensatz sehr gut ausgeprägt. Jira agile bringt Scrum- und Kanbanfunktionen mit.

ALM Platform: Die Cloud-Lösung „ALM Platform“ von Rally zeichnet sich durch ein sehr hohes Anpassungspotential, viele Integrations-, Import- und Exportfunktionen und eine sehr gut Plattform aus. Zusätzlich vereint das Tool zahlreiche Scrum- und Kanban-Funktionalitäten in einem Tool.

KADOS: KADOS ist ein gutes Open Source Tool für kleine Projekte und Teams. Von Vorteil ist die kostenfreie Version, die individualisiert werden kann.

AgileFant: AgileFant unterstützt die Scrum Prinzipien erfolgreich, Kanban jedoch gar nicht. Die Implementierung ist einfach und es gibt eine ausgebaute Community, jedoch mangelt es an Anpassungsmöglichkeiten und Kollaborations-Funktionen.

IceScrum: Das Open Source Tool IceScrum besitzt viele Scrum und auch ein paar Kanban Merkmale. Jedoch eignet es sich eher für kleinere Teams und der Datenaustausch mit anderen Tools wird kaum unterstützt.

5 Fazit

Ziel dieser Arbeit war es einen Vergleich zu den sich momentan am Markt befindlichen Tools zur leichtgewichtigen Softwareentwicklung im kommerziellen und Open Source Bereich durchzuführen und eine Bewertungsgrundlage zur Auswahl eines auf die Anforderungen des jeweiligen Unternehmens passenden Tools zu schaffen. Zur Unterstützung der leichtgewichtigen Softwareentwicklung werden verschiedene Frame-works und Methoden genutzt. Eines der wichtigsten Frameworks ist Scrum, während bei den Methoden Kanban eine bedeutende Rolle spielt.

Bei dem agilen Framework Scrum liegt der Hauptfokus auf dem Team und nicht den Prozessen. Aus diesem Grund finden iterative Teammeetings statt, sogenannte Sprints. Am Ende jedes Sprints muss ein fertiges Teilprodukt des Gesamtproduktes geliefert werden, dadurch werden eine bessere Qualität, Dynamik und Flexibilität garantiert. Neben Sprints finden noch weitere regelmäßige Meetings statt, in denen auf Änderungen und neue Anforderungen reagiert wird, welche in den folgenden Sprints umgesetzt werden.[58] Auf Änderungen der Umwelt oder neue Anforderungen bzw. Wünsche des Kunden, die während des Projekts auftreten, kann schnell und effizient reagiert werden. Der Fokus liegt auf dem Ziel und dem effizienten Erreichen des gewünschten und geforderten Ergebnisses bzw. Produkts. Der Kunde wird in den Verlauf des agilen Projekts involviert und hat die Möglichkeit, Feedback zu geben und weitere Anforderungen zu äußern. Im Projektteam ist Transparenz sehr wichtig. Diese ermöglicht es Probleme zusammen und schnell zu lösen und diese in Zukunft zu vermeiden. Viele größere Softwarehersteller bieten Tools für den Einsatz von Scrum-Methoden an. Dazu gehören z.B. Atlassian (JIRA Agile), IBM (Agile Development), HP (Agile Manager) und Rally (ALM Platform).

Bei Kanban handelt es sich im Gegensatz zu Scrum um eine unterstützende Methode, bei der Aufgaben nach festgelegten Regeln abgearbeitet und kontrolliert werden. Hierbei liegt der Hauptfokus auf der Sicherstellung eines fortlaufenden und durchgängigen Arbeitsflusses (Workflows), d.h. dass der Fokus nicht auf dem Team, sondern auf dem Prozess liegt.[59] Hier werden die Aufgaben visualisiert, mit dem Ziel den Durchsatz zu erhöhen. Die Aufgaben werden in Form von Tickets auf einem „Board“ erfasst. Dadurch kann der aktuelle Status des Projekts dargestellt werden. Für die Umsetzung der Kanban-Methoden stellen größere Softwarehersteller Tools bereit, wie z.B. Leankit und Swiftkanban.

Insgesamt wurden 9 Produkte, davon sechs kommerzielle und drei aus dem Open Source Bereich, in den Kategorien Scrum, Kanban, Kollaboration, Implementierung, Plattform, Schnittstel-

[58] Vgl. Dräther, R./Koschek, H./Sahling, C. (2013), S. 13f.
[59] Vgl. Sahota, M. (2012), S.11-14

len und Customization mithilfe eines im Vornherein erstellten Kriterienkatalogs analysiert und anhand eines Gewichtungssystems entsprechend bewertet. Für jede Kategorie werden mehrere detaillierte Unterpunkte (Kriterien) aufgeführt, welche für jedes Tool mithilfe einer Punkteskala einzeln bewertet werden. Durch die Arbeit mit einem Excel-Formular, in dem die Kriterien und Bewertungen notiert werden, können einzelne Punkte außerdem stärker gewichtet werden als andere. Ist einem Team zum Beispiel wichtig, dass ein Tool besonders Scrum unterstützt, kann dieser Kategorie eine höhere Bedeutung bei der Gesamtbewertung gegeben werden.

Basierend auf der durchgeführten Bewertung und dem Tool-Vergleich lässt sich eine Tool-Empfehlung abgeben. Im Bereich der kommerziellen Tools hat sich für Projekte mit Scrum-Fokus das Tool „Rally“ durchgesetzt, welches in allen Kategorien punkten konnte. Des Weiteren sind jedoch auch die Anwendung von IBM sowie das Tool „JIRA agile“ für Scrum-Ansätze zu empfehlen. Für Kanban-Projekte hat sich ebenfalls das Tool „Rally“ durch hervorragende Funktionalitäten hervorgetan. Ebenfalls sehr gut abgeschnitten hat das Produkt „Leankit“.

Die kommerziellen Produkte haben im Vergleich mit den Open Source Tools in Bezug auf ihre Funktionsvielfalt durchweg besser abgeschnitten. Unter den Open Source Produkten wiesen die Anwendungen „AgileFant“ und „IceScrum“ mehr Funktionen auf. Allerdings ist das Produkt „KA-DOS“ die einzige komplett kostenfreie Anwendung der getesteten Lösungen.

Die Produktvielfalt am Markt ist sehr groß und schlussendlich lässt sich sagen, dass die Wahl des „besten“ Produktes immer eng verbunden ist mit den Projektmerkmalen und den gegebenen Voraussetzungen. Es gibt keinen speziellen „Sieger“ der aus der Marktstudie hervorgegangen ist, sondern vielmehr ein eher geeignetes Produkt basierend auf den Anforderungen des jeweiligen Teams.

Anhang

Anhangsverzeichnis

Anhang 1: Excel-Sheets mit Kriterienkatalog und Tool-Vergleich

Kriterien		Swiftkanban			Leankit			Agilefant		
Funktionale Anforderungen	Gewichtung	Begründung	Bewertung	Score	Begründung	Bewertung	Score	Begründung	Bewertung	Score
Scrum	40,00%			7,555556			1,111111			10,2222
Backlog-Erstellung	5	Karten können	7	35		0	0	Produkte, Proj	8	40
Backlog-Items	5	gute Backlog-S	7	35		0	0	Story tree	8	40
Task Board	5	übersichtliches	5	25	Scrum Board av	5	25	Task queue un	8	40
Release Burndown Chart	5	RBC vorhanden	5	25		0	0		0	0
Sprint Burndown Chart	5	SPC vorhanden	5	25		0	0	Zeigt Iterations	7	35
Velocity Ermittlung & Chart	5		0	0		0	0		0	0
Sprint Planning	5	vorhanden	5	25		0	0	Sprints = Iterati	8	40
Weitere Reportings	5	keine	0	0	keine	0	0	Burnup Chart, T	7	35
Zusatzfunktionen	5	keine	0	0	keine	0	0	keine	0	0
Kanban	40,00%			12,85714			22,62857			1,42857
Kanban Boardgestaltung	5	Übersichtliches	7	35	Verschiedene I	8	64	keine spezielle	0	0
Geschachtelte Prozesse	5	Mehrere Projel	8	40	Verlinkung der	5	25	Hierarchie der	5	25
Cumulative Flow Diagram	5	Vorhanden	9	45	Leicht verständ	9	81		0	0
WIP Organisation	5	WIP Limit	5	25	WIP Maximieru	9	81		0	0
Kartengestaltung	5	Einfach Karten,	8	40	Viele Kartende	9	81		0	0
Weitere Reportings	5	Weitere Repor	6	30	Efficiency, cycl	8	64	keine	0	0
Zusatzfunktionen	5	Karten verschw	2	10	keine	0	0	keine	0	0
Kollaborationsfunktionen	15,00%			5,25			4,8			2,1
Online Community	5	Blog und online	7	35	User-Communi	9	45	Online Commu	8	40
Nachrichtensystem	5	Chatfunktion	8	40	Benachrichtigu	2	10	Chat Funktion I	6	30
Teamkalender	5	Kalenderansicl	8	40	Kalendersicht e	9	45		0	0
Scalability	5	gut	5	25	gut	5	25		0	0
Feedback	5	Chatfunktion	7	35	Kommentare b	7	35	keine Feedbacl	0	0
Zusatzfunktionen	5,00%			0			0			0
	20,00%									
Total Funktionale Anforderungen	100,00%			25,6627			28,53968			13,7508

Abb. 1: Kriterienkatalog für SwiftKanban, LeanKit, AgileFant

Nichtfunktionale Anforderungen	Gewichtung	Begründung	Bewertung	Score	Begründung	Bewertung	Score	Begründung	Bewertung	Score
Implementierung	25,00%			7,291667			8,125			8,33333
Kaufpreis	5	15€ für Enterpr	7	35	19€ für Portfoli	6	30	20$ für Enterpr	6	30
laufende Kosten	5	monatliche Kos	5	25	monatliche Kos	5	25	monatliche Kos	5	25
Installation	5	Einloggen onlir	9	45	Einloggen onlir	9	45	Einloggen onlir	9	45
Sytemanforderungen	5	Browser	9	45	Browser	9	45	Browser	9	45
Hosting/Licensing Optionen	5	keine Alternati	0	0	Private cloud	5	25	Hosting auf eig	6	30
Instandhaltung	5	einfach	5	25	einfach	5	25	einfach	5	25
Plattform	25,00%			6,607143			8,571429			7,85714
Suchfunktion	5	Suchfunktion d	6	30	Suchfunktion n	8	40	Suchfunktion ü	5	25
Mobile Nutzung/Interface	5	Mobil über Bro	5	25	Mobil einfach r	8	40	Mobile App be	5	25
Sicherheit	5	Gut	5	25	Gut, zusätzliche	8	40	Zusätzliche Sicl	8	40
Stabilität	5	Gut	5	25	Gut	5	25	Gut	5	25
Verbreitungsgrad	5	moderat	2	10	moderat	2	10	weit verbreitet	6	30
Support	5	Live Chat verfü	7	35	Online Support	8	40	Online Support	8	40
Trainings	5	Guide und Hilfe	7	35	Mail von execu	9	45	Online tutorial	7	35
Schnittstellen	25,00%			5			5,416667			5
Erweiterbarkeit/Integration	5	Integration übe	7	35	Verbindung zu	8	40	kaum Möglichk	0	0
Integration mit Entwicklungsumgebung	5	k.A.	0	0	k.A.	0	0	Verbindung de	7	35
Import/Export Funktion	5	Import von Kar	5	25	Karten können	5	25	Excel Export, E)	5	25
Customization	25,00%			6,25			6,5625			4,0625
Benutzerverwaltung	5	Benutzer könn	6	30	Benutzer könn	5	25	Wenig Nutzerd	2	10
intuitive Bedienung	5	Leicht bedienb	7	35	Drag&Drop Sys	8	40	Drag&Drop	5	25
Anpassbarkeit/custom fields	5	Personalisierba	6	30	Personalisierba	7	35	Custom dashbc	5	25
Sprache	5	Englisch	1	5	Englisch	1	5	Englisch	1	5
Total Nichtfunktionale Anforderungen	100,00%			25,14881			28,6756			25,253
Gesamtscore				50,81151			57,21528			39,0038
Platzierung				2			1			3

Abb. 2: Kriterienkatalog (2) für SwiftKanban, LeanKit, AgileFant

Kriterien		IBM - Agile Development			HP - Agile Manager			KADOS		
Funktionale Anforderungen	**Gewichtung**	**Begründung**	**Bewertung**	**Score**	**Begründung**	**Bewertung**	**Score**	**Begründung**	**Bewertung**	**Score**
Scrum	40,00%			8,444444			8			7,77778
✔ Backlog-Erstellung	5	Product- und R	7	35	Möglich	5	25	Vorhanden	5	25
✔ Backlog-Items	5	Stack-ranking d	7	35	Vorhanden	5	25	Kategorisierun	6	30
✔ Task Board	5	Unterteilt in to	7	35	Einfache Taskb	5	25	Task Status boa	7	35
✔ Release Burndown Chart	5	Vorhanden	5	25	Structure relea	6	30	Realse Hirachie	5	25
✔ Sprint Burndown Chart	5	Vorhanden	5	25	User Stories -->	5	25	Sprint Hirachie	7	35
✔ Velocity Ermittlung & Chart	5	Balken zum "Pr	7	35	Automatisches	5	25		0	0
✔ Sprint Planning	5		0	0	Planung von Us	5	25	Mit Hirchrie de	5	25
✔ Weitere Reportings	5	keine	0	0	keine	0	0	keine	0	0
✔ Zusatzfunktionen	5	keine	0	0	keine	0	0	keine	0	0
Kanban	40,00%			12,28571			18,57143			3,14286
✔ Kanban Boardgestaltung	5	Vorhanden	5	25	Vorhanden	5	25	Bestehend aus	6	30
✔ Geschachtelte Prozesse	5		8	40		5	25	Hierarchie der	5	25
✔ Cumulative Flow Diagram	5	Cumulative Val	8	40	Vorhanden (un	7	49		0	0
✔ WIP Organisation	5	WIP Limits	6	30		9	81		0	0
✔ Kartengestaltung	5		8	40		9	81		0	0
✔ Weitere Reportings	5		6	30		8	64		0	0
✔ Zusatzfunktionen	5		2	10		0	0		0	0
Kollaborationsfunktionen	15,00%			4,65			0,45			0,45
✔ Online Community	5	enterprise soci	9	45	Ausbaufähig, k	3	15	Forum, aber ka	3	15
✔ Nachrichtensystem	5	Chat mit Anwe	8	40		0	0		0	0
✔ Scalability	5		0	0		0			0	0
✔ Teamkalender	5	Scheduled Abs	7	35		0	0		0	0
✔ Feedback	5	Review Funktic	7	35		0	0		0	0
Zusatzfunktionen	5,00%			0			0			0
	20,00%									
Total Funktionale Anforderungen	100,00%			**25,38016**			**27,02143**			**11,3706**

Abb. 3: Kriterienkatalog für IBM, HP und KADOS

Nichtfunktionale Anforderungen	Gewichtung	Begründung	Bewertung	Score	Begründung	Bewertung	Score	Begründung	Bewertung	Score
Implementierung	25,00%			2,291667			3,958333			7,08333
Kaufpreis	5	214 EUR für Sin	4	20	ab 39 $ / Monat	6	30	Kostenlos	10	50
laufende Kosten	5		0	0	ab 39 $ / Monat	5	25	Kostenlos	5	25
Installation	5	Sehr komplex	3	15	einfache Instal	8	40	Einloggen onlir	9	45
Sytemanforderungen	5	Komplex	4	20		0	0	web server wit	5	25
Hosting/Licensing Optionen	5		0	0		0	0	Self-hosted	5	25
Instandhaltung	5		0	0		0	0		0	0
Plattform	25,00%			6,071429			5,178571			3,03571
Suchfunktion	5	Quick Search ui	7	35		0	0		0	0
Mobile Nutzung/Interface	5		0	0		0	0		0	0
Sicherheit	5		0	0	ISO/IEC 27001:2	8	40	Sicher, da selbs	8	40
Stabilität	5	Lack of stability	4	20	SLA - Availabili	6	30		0	0
Verbreitungsgrad	5	Marktführer	8	40	Challenger	7	35		3	15
Support	5	Online Support	8	40	24x7x365 (Tele	8	40	forum on Sourc	6	30
Trainings	5	Web based Tra	7	35		0	0		0	0
Schnittstellen	25,00%			0			1,666667			0
Erweiterbarkeit/Integration	5		0	0	Nur Windows u	4	20		0	0
Integration mit Entwicklungsumgebung	5		0	0		0	0		0	0
Import/Export Funktion	5		0	0		0	0		0	0
Customization	25,00%			2,8125			9,6875			2,1875
Benutzerverwaltung	5		0	0	Verschiedende	7	35		0	0
intuitive Bedienung	5		0	0	Einfache Benut	8	40		0	0
Anpassbarkeit/custom fields	5		0	0	Anpassbare Re	8	40		0	0
Sprache	5	Versch. Sprach	9	45	10 verschiende	8	40	French, English	7	35
Total Nichtfunktionale Anforderungen	100,00%			11,1756			20,49107			12,3065
Gesamtscore				36,55575			47,5125			23,6772
Platzierung				2			1			3

Abb. 4: Kriterienkatalog (2) für IBM, HP und KADOS

Kriterien		JIRA Agile			ALM Platform			IceScrum		
Funktionale Anforderungen	**Gewichtung**	**Begründung**	**Bewertung**	**Score**	**Begründung**	**Bewertung**	**Score**	**Begründung**	**Bewertung**	**Score**
Scrum	40,00%			9,333333			10,88889			10,2222
Backlog-Erstellung	5	multi-select rai	6	30		5	25		5	25
Backlog-Items	5	User Story Man	5	25	User Story Man	5	25	User Story Man	5	25
Task Board	5	Swimlanes auf	6	30	iteration overv	6	30		5	25
Release Burndown Chart	5		5	25		5	25		5	25
Sprint Burndown Chart	5		5	25		5	25		5	25
Velocity Ermittlung & Chart	5		5	25		5	25		5	25
Sprint Planning	5		5	25		6	30		5	25
Weitere Reportings	5	keine	0	0	Performance R	7	35	Sprint und Burr	5	25
Zusatzfunktionen	5	Workflows, Act	5	25	Activity Boards	5	25	Filter, Product	6	30
Kanban	40,00%			3,714286			8,4			2,85714
Kanban Boardgestaltung	5	Kanban Board	5	25	Kanban-Boards	6	36		0	0
Geschachtelte Prozesse	5		3	15		0	0		0	0
Cumulative Flow Diagram	5		5	25		5	25		5	25
WIP Organisation	5		0	0		0	0		5	25
Kartengestaltung	5		0	0		5	25		0	0
Weitere Reportings	5	keine	0	0	Performance R	5	25	keine	0	0
Zusatzfunktionen	5	Swimlanes, Wc	0	0	Defect Charts, I	6	36	keine	0	0
Kollaborationsfunktionen	15,00%			3,45			1,8			3
Online Community	5		0	0		6	30	Fragen werden	6	30
Nachrichtensystem	5	HipChat = muss	4	20	real-time feedl	6	30		5	25
Teamkalender	5		6	30		0	0		0	0
Scalability	5	multi-project L	6	30	mutli-projekt u	8		Mehr für kleine	4	20
Feedback	5	Anmerkungen	7	35		0	0		5	25
Zusatzfunktionen	5,00%			0			0,3125			0
	20,00%									
Total Funktionale Anforderungen	100,00%			**16,49762**			**21,40139**			**16,0794**

Abb. 5: Kriterienkatalog für Jira Agile, ALM Platform und IceScrum

Nichtfunktionale Anforderungen	Gewichtung	Begründung	Bewertung	Score	Begründung	Bewertung	Score	Begründung	Bewertung	Score
Implementierung	25,00%			3,958333			3,125			3,95833
Kaufpreis	5	10 Benutzer = $	5	25	muss angefragt	5	25	Jahreslizenz fü	4	20
laufende Kosten	5		0	0		0	0		0	0
Installation	5		0	0	Keine Installati	5	25		5	25
Sytemanforderungen	5	HTML5	7	35		0	0	Java Container,	5	25
Hosting/Licensing Optionen	5	Cloud/SaaS ode	7	35	Cloud-Based (S	5	25	Server- und Clo	5	25
Instandhaltung	5		0	0		0	0		0	0
Plattform	25,00%			6,25			6,428571			6,25
Suchfunktion	5	Um Items im Ba	8	40		6	30		6	30
Mobile Nutzung/Interface	5		5	25		0	0	in Bearbeitung	4	20
Sicherheit	5		7	35		5	25		4	20
Stabilität	5		0	0		5	25	Häufig Problem	4	20
Verbreitungsgrad	5	ca. 20.000 Unte	5	25		7	35	3000 Download	7	35
Support	5		7	35	Support für Pro	7	35	on demand sup	5	25
Trainings	5	Nicht sehr viele	3	15	Tutorials und Ir	6	30	Youtube Trainii	5	25
Schnittstellen	25,00%			7,5			7,5			5,83333
Erweiterbarkeit/Integration	5	Tickets von anc	9	45	vorgefertige Co	6	30		5	25
Integration mit Entwicklungsumgebung	5	Erweiterbar mi	6	30	Integration zu \	6	30	Integration mit	6	30
Import/Export Funktion	5		3	15	Daten von vorh	6	30	Daten können (	3	15
Customization	25,00%			8,75			8,75			5,9375
Benutzerverwaltung	5		5	25		5	25		6	30
intuitive Bedienung	5		7	35		9	45		6	30
Anpassbarkeit/custom fields	5	Eigene Filter si	7	35	Es kann so gut \	9	45		2	10
Sprache	5	14 Sprachen	9	45	Englisch	5	25	Englisch	5	25
Total Nichtfunktionale Anforderungen	100,00%			26,45833			25,80357			21,9792
Gesamtscore				42,95595			47,20496			38,0585
Platzierung				2			1			3

Abb. 6: Kriterienkatalog (2) für Jira Agile, ALM Platform und IceScrum

Quellenverzeichnis

Literaturverzeichnis

Dräther, R./Koschek, H./Sahling, C. (2013): Scrum. Kurz & gut. Köln: O'Reilly Verlag.

Epping, T. (2011): Kanban für die Softwareentwicklung. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg.

Ladas, C. (2008): Scrumban. And other essays on Kanban System for Lean Software develoment. Saettle, WA: Modus Cooperandi Press.

Rubin, K. S. (2014): Essential Scrum. Umfassendes Scrum-Wissen aus der Praxis. Heidelberg: mitp.

Sahota, M. (2012): Agile adoption and transformation survival guide. o. O: Lulu Com.

Arbeiten

Hense, A. (2012): Evaluierung agiler Vorgehensmodelle in der Softwareentwicklung. Bachelorarbeit.

http://edoc.sub.uni-hamburg.de/haw/volltexte/2012/1588/pdf/Alex_Hense_Evaluierung_ agiler_Vorgehensmodelle.pdf, Hamburg: Hochschule für Angewandte Wissenschaften Hamburg. Abruf: 05.12.2014.

Internetquellen

Agilefant Ltd. (2015a): User guide - Agilefant. http://agilefant.com/support/user-guide/, Abruf: 08.01.2014.

Agilefant Ltd. (2015b): Features - Agilefant. http://agilefant.com/features, Abruf: 07.01.2015.

Atlassian Inc. (2015a): JIRA - Software für Problempunkt- und Projekt-Tracking. https://de.atlassian.com/software/jira, Abruf: 02.12.2014.

Atlassian Inc. (2015b): Atlassian: A Leader in Gartner's 2013 Magic Quadrant for Application Development Life Cycle Management. https://de.atlassian.com/gartner/, Abruf: 07.01.2015.

Atlassian Inc. (2015c): JIRA Agile | Atlassian. https://www.atlassian.com/software/jira/agile, Abruf: 01.12.2014.

Beck u.a. (2001): Manifesto for Agile Software Development. http://agilemanifesto.org/, Abruf: 01.12.2014.

Bock, J. (2013): Agiles Projektmanagement: Scrum, Kanban und Scrumbuts im Einsatz. t3n Magazin. http://t3n.de/magazin/praxisbericht-scrum-kanban-scrumbuts-agiles-232822/, Abruf: 12.12.2014.

Digité, Inc. (2015a): Features and Product Overview | SwiftKanban. http://www.swiftkanban.com/kanban-scrum-scrumban-features.html?utm_expid=64655604-10.wxmdc6ePS024QuGNBs2t7Q.0&utm_referrer=http%3A%2F%2Fwww.swiftkanban.com%2F, Abruf: 01.12.2014.

Digité, Inc. (2015b): What is Kanban? | SwiftKanban Homepage. http://www.swiftkanban.com/kanban/1-what-is-kanban.html, Abruf: 02.12.2014.

Duggan, J./Murphy, T. E. (2012): Magic Quadrant for Application Life Cycle Management. http://www.techostan.com/docs/quadrant.pdf, Abruf: 05.01.2015.

Erickson, K. (2013): A Comparative Look at Top Agile Tools. https://www.captechconsulting.com/blog/kevin-erickson/comparative-look-top-agile-tools,

Abruf: 10.12.2014.

Grant, T. (2012): The Forrester Wave™: Application Life- Cycle Management, Q4 2012. https://www.rallydev.com/sites/default/files/the-forrester-wave-application-life-cycle-management-2012.pdf, Abruf: 07.01.2015.

heise Developer (2010): Rally übernimmt Kanban- Werkzeug. http://www.heise.de/developer/meldung/Rally-uebernimmt-Kanban-Werkzeug-978238.html, Abruf: 05.12.2014.

Hewlett-Packard Development Company, L.P. (2015): Guide to Build and Source Code Analytics in Agile Manager. http://h10124.www1.hp.com/campaigns/IT_Experts/pages/Agile_Manager.html, Abruf: 02.12.2014.

Holdorf, C. (2011): Agile Tool Selection. http://de.slideshare.net/choldorf/agile-tools, Abruf: 17.01.2015.

IBM (2014): Rational Team Concert. https://jazz.net/products/rational-team-concert/, Abruf: 01.12.2014.

it-agile GmbH (o. J.): Scrum. http://www.it-agile.de/wissen/methoden/scrum/, Abruf: 06.12.2014.

Jokelainen, R./Pekkanen, P. (o. J.): Agilefant - Agile Backlog Product and Project Management. http://www.methodsandtools.com/tools/tools.php?agilefant, Abruf: 07.01.2015.

Kagilum SAS (2012): Kagilum, professional services for iceScrum tool. https://www.kagilum.com/, Abruf: 07.01.2015.

LeanKit Inc. (2014a): Product | LeanKit. http://leankit.com/product/, Abruf: 02.12.2014.

LeanKit Inc. (2014b): Board Analytics. https://support.leankit.com/hc/en-us/articles/204413593-Board-Analytics, Abruf: 10.12.1014.

Leffingwell u.a. (2015): Scaled Agile Framework. http://www.scaledagileframework.com/, Abruf: 07.01.2015.

Marmotte Technologies (o. J.): ScreenShots | KADOS. http://www.kados.info/en/?page_id=18, Abruf: 08.01.2015.

Mountain Goat Software, LLC (o. J.): userStories - All products. http://www.userstories.com/products, Abruf: 20.01.2015.

Peterson, D. (2014): What is Kanban? http://kanbanblog.com/explained/, Abruf: 01.12.2014.

Quack, K. (2014): Kanban erobert die IT-Branche: Nach Scrum kommt nun Kanban. http://www.computerwoche.de/a/nach-scrum-kommt-nun-kanban,3071288, Abruf: 05.01.2015.

Rally Software Development Corp. (2015a): Rally Editions | Rally Software. https://www.rallydev.com/platform-products/rally-editions, Abruf: 01.12.2014.

Rally Software Development Corp. (2015b): Rally Platform | Rally Software. https://www.rallydev.com/platform-products/rally-platform, Abruf: 01.12.2014.

Rally Software Development Corp. (2015c): Rally Demo Center | Rally Software. https://www.rallydev.com/rally-demo-center, Abruf: 01.12.2014.

Rally Software Development Corp. (2015d): Rally Software Delivers Multi-process Agile Support for Scrum, Kanban and High Assurance Methods | Rally Software. https://www.rallydev.com/about/rally-software-delivers-multi-process-agile-support-scrum-kanban-and-high-assurance-methods, Abruf: 01.12.2014.

Roock, A. (o. J.): Software-Kanban - eine Einführung. https://www.projektmagazin.de/artikel/software-kanban-eine-einfuehrung_996703, Abruf: 20.12.2014.

Scrum-Master.de (2014): Scrum - auf einer Seite erklärt. http://scrum-master.de/Was_ist_Scrum/Scrum_auf_einer_Seite_erklaert, Abruf: 06.12.2014.

Tan, E./Weiss, J. (2004): Klassische vs. agile Methoden der Softwareentwicklung. http://www.codesco.com/fileadmin/bilder_grafiken/Sonstiges/KlassischVSagil_Ausarbeitung.pdf, Abruf: 05.12.2014.

The Standish Group International, Inc (2011): Chaos Manifesto. The Laws of CHAOS and the CHAOS 100 Best PM Practices. http://www.versionone.com/assets/img/files/ChaosManifest_2011.pdf, Abruf: 05.12.2014.

Twitter (2014): icescrum (@icescrum) | Twitter. https://twitter.com/icescrum, Abruf: 08.01.2015.